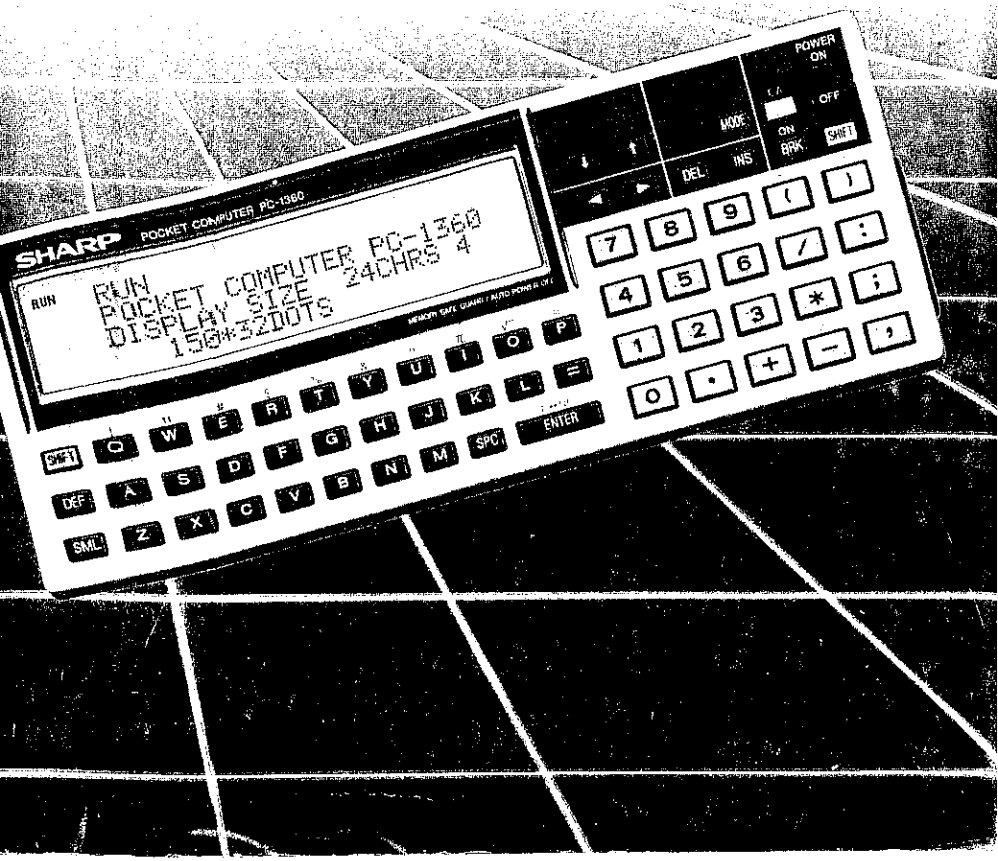
SHARP®

SCHENCOMPUTER

MODELL

PC-1360

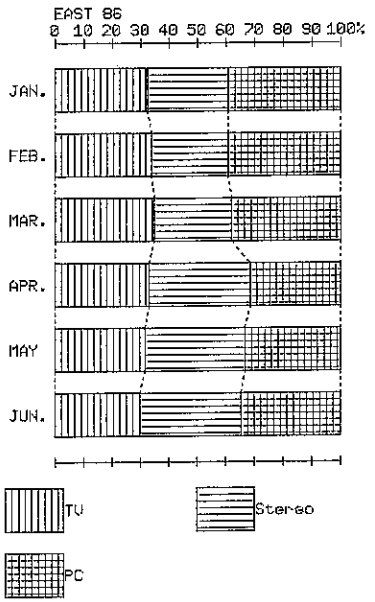
BEDIENUNGSANLEITUNG



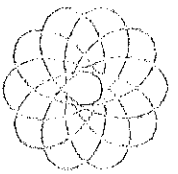
FARBDRUCK

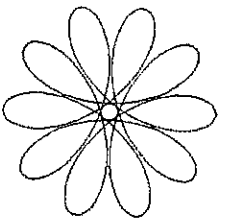
■ Vergleichende Band-Grafik
(Siehe Seite 343)

	TV	Stereo	PC
JAN.	30	26	36
FEB.	28	22	32
MAR.	26	20	28
APR.	34	36	32
MAY	29	32	30
JUN.	25	30	29

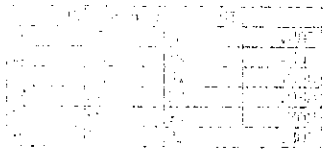


■ Runde Grafik
(Siehe Seite 359)





- Maßstab um 55% verkleinert



EINLEITUNG

Willkommen in der Welt der SHARP-Besitzer!

Nur wenige Industriezweige der Welt können heute Schritt halten mit dem schnellen Wachstum und dem technischen Fortschritt im Bereich der Personalcomputer. Computer, die noch vor kurzer Zeit einen ganzen Saal gefüllt hätten, deren Programmierung einen akademischen Abschluß erfordert und die Tausende von Dollars gekostet hätten, passen heute in Ihre Handfläche, sind einfach zu programmieren und kosten so wenig, daß fast jeder sie sich leisten kann.

Ihr neuer SHARP PC-1360 wurde entwickelt, um Ihnen den allerneuesten Stand dieser Computerrevolution zugänglich zu machen und verfügt über eine Vielzahl fortschrittlichster Fähigkeiten:

- SPEICHERSCHUTZ - der PC-1360 merkt sich gespeicherte Programme und Variablen, selbst wenn Sie ihn abschalten.
- Stromversorgung durch Batterien, ideal für ein wirklich portables System.
- AUTOMATISCHE ABSCHALTUNG, die die Batterien schützt, indem die Stromversorgung abgeschaltet wird, wenn innerhalb eines bestimmten Zeitraums keine Operation durchgeführt wird.
- Eine erweiterte Version des BASIC, die formatierte Datenausgabe ermöglicht, weiterhin zweidimensionale Felder, variable Zeichenfolgen-Länge, Programmverkettung und viele andere fortschrittliche Arbeitsmöglichkeiten.
- Als Option das Drucker/Kassettenrecorder-Interface (Modell CE-126P), mit dem Sie dauerhafte Kopien von Programmen und Daten erstellen können. Das Kassetten-Interface erlaubt es Ihnen, einen Kassettenrecorder anzuschließen, um Programme und Daten auf Kassette zu speichern.
- Ein serielles E/A-Interface, das eine direkte Verbindung von Computer zu Computer erlaubt und Anpassung für den vielseitigeren Farbdrucker CE-140P und den Farbplotter CE-515P.
- Speicherflexibilität mit Hilfe von zwei RAM-Einbauplätzen.

Wir gratulieren Ihnen zu Ihrem Eintritt in eine aufregende und faszinierer
de neue Welt. Wir sind sicher, daß Sie Ihre neue Errungenschaft schon bal
als eine der klügsten betrachten werden, die Sie je getätigt haben. De
SHARP PC-1360 ist ein hilfreiches Werkzeug, das entwickelt wurde, um Ih
ren spezifischen mathematischen, wissenschaftlichen, technischen, kauf
männischen und persönlichen Computerbedürfnissen gerecht zu werden
Mit dem SHARP PC-1360 können Sie JETZT damit anfangen, die Lösun
gen zu finden, die Sie morgen brauchen werden!

INHALTSVERZEICHNIS

EINLEITUNG Seite

1 WIE MAN DIESES HANDBUCH BENUTZT 1

2 EINFÜHRUNG IN DEN PC-1360 3

Beschreibung der Tastatur 4
Beschreibung des Zeilendisplays 8
ALL RESET-Schalter 10
Kontrastregulierung 11
Batteriewechsel 12
Einsetzen der Batterien 12
Vor dem Gebrauch des PC-1360 16

3 DER GEBRAUCH DES PC-1360 ALS RECHNER 19

Einschalten des Computers 19
Ausschalten des Computers 19
Automatische Abschaltung 20
Einige hilfreiche Hinweise 20
Einfache Rechenoperationen 21
Wiederabruf von Eingaben 23
Fehler 27
Verkettung von Rechenoperationen 28
Negative Zahlen 30
Komplexere Rechenoperationen und Klammerung 31
Der Gebrauch von Variablen in Rechenoperationen 32
Mehrere Rechenschritte 34
Wissenschaftliche Notation 35
Grenzen 36
Werte-Schablone 36
Maximale Rechenlänge 38
Wissenschaftliche Rechenoperationen 38
Rangfolge beim Rechnen ohne Programmunterstützung 41
Ausdruck von Rechenergebnissen 42

4 BEGRIFFE UND AUSDRÜCKE DES BASIC 44

Zeichenfolgen-Konstanten 44
Hexadezimalzahlen 45
Variablen 45
Vorgegebene Variablen 47
Einfache Variablen 48
Feld-Variablen 49
Variablen der Form A( ) 53
Ausdrücke 56
Numerische Operatoren 56
Zeichenfolgen-Ausdrücke 57
Verhältnis-Ausdrücke 57
Logische Ausdrücke 58
Klammerung und Vorrang der Operatoren 61
RUN-Modus 61
Funktionen 62

5 PROGRAMMIEREN DES PC-1360 63

Programme 63
BASIC-Anweisungen 63
Zeilennummern 64
BASIC-Befehle 64
BASIC-Kommandos 65
Modi 65
Der Anfang des Programmierens mit dem PC-1360 66
Beispiel 1 — Eingabe und Ablauf eines Programms 67
Beispiel 2 — Aufbereitung eines Programms 67
Beispiel 3 — Gebrauch von Variablen beim Programmieren 70
Beispiel 4 — Komplexere Programmierung 74
Das Speichern von Programmen im PC-1360 75
Grafikfunktionen auf dem Bildschirm 76

6 ABKÜRZUNGEN 82

Die DEF-Taste und benannte Programme 82
RESERVE-Modus 83
Schablonen 86

7 ZUSÄTZLICHER PERIPHERIEGERÄTE 87

CE-126P Thermaldrucker/Kassetteninterface 87
CE-140P Mehrfarben Matrixdrucker 88
CE-515P1516P Farb-Plotter/Drucker 88
CE-130T Pegelumsetzer 89
CE-124 Kassetteninterface 89
Verwendung des CE-126P drucker/Kassetteninterface 90
Anmerkungen zum Tonband 96
Gebrauch von Farbdrucker 97
Serielle E/A Funktion 98

8 GEBRAUCH DER RAM-KARTEN 102

Installation der RAM-Karte 102
Entnahme der RAM-Karte 103
Einsatz der RAM-Karte 104
Vorsichtsmaßnahmen beim Einsatz der RAM-Karten 106
Das Kopieren von RAM-Karten 107
Gebrauch von RAM-Karten anderer Computer 108

9 COMPUTER-ANWEISUNGEN DES PC-1360 110

Kommandos 110
Befehle 110
Funktionen 110

10 PANNENHILFE 303

Bedienung des Geräts 303
Fehlersuche im BASIC 304

11 INSTANDHALTUNG DES PC-1360 310

ANHANG 311

Anhang A: Fehlermeldungen 311
Anhang B: Zeichencode-Tabelle 314
Anhang C: Formatierung der Datenausgabe 318
Anhang D: Bewertung von Ausdrücken und Vorrang von Operatoren 323
Anhang E: Tastenfunktionen im BASIC 326
Anhang F: Symbole des E/A-Anschlusses 332
Anhang G: Technische Daten 334
Anhang H: Benutzung von Programmen auf dem PC-1360, die für andere SHARP Modelle entwickelt wurden 337

PROGRAMMBEISPIELE 339

INDEX 367

1 WIE MAN DIESES HANDBUCH BENUTZT

Dieses Handbuch soll Sie einführen in die Fähigkeiten und Charakteristika Ihres PC-1360 und es soll Ihnen als wertvolles Nachschlagewerk dienen. Ob sie ein "Einsteiger" sind oder ob die Computer zu Ihren "alten Freunden" zählen, Sie sollten sich mit dem PC-1360 vertraut machen, indem Sie die Kapitel 2 bis 6 lesen und durcharbeiten.

- Kapitel 2 beschreibt das äußere Erscheinungsbild des PC-1360.
- Kapitel 3 erklärt den Gebrauch des PC-1360 als wissenschaftlicher Taschenrechner.
- Kapitel 4 erklärt einige Ausdrücke und Begriffe, die für die Programmierung in BASIC von Bedeutung sind, und es erläutert Ihnen die spezielle Bedeutung dieser Begriffe für die Arbeit mit dem PC-1360.
- Kapitel 5 dient als Einführung in die BASIC-Programmierung mit dem PC-1360. Es zeigt, wie man Programme eingibt, korrigiert und abarbeiten kann.
- Kapitel 6 behandelt einige Tips und Kniffe, mit denen Sie sich die Arbeit mit Ihrem neuen Computer weiter vereinfachen und noch mehr Spaß damit haben können.

Erfahrene BASIC-Programmierer können dann noch Kapitel 9 durchlesen, um die spezifischen Züge des BASIC im PC-1360 kennenzulernen. Da sich alle BASIC-Dialekte geringfügig unterscheiden, empfiehlt es sich, dieses Kapitel mindestens einmal durchzulesen, bevor man mit dem ernsthaften Programmieren beginnt.

Kapitel 9 ist ein Nachschlageteil, der alle Befehle, Kommandos und Funktionen des BASIC enthält.

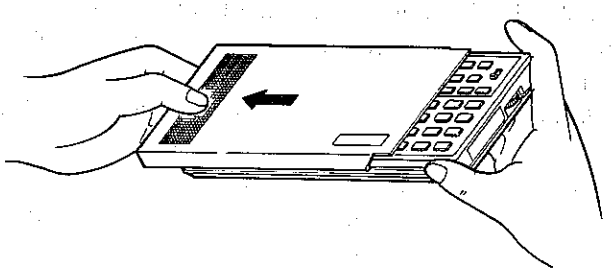
Wenn Sie bislang noch nicht in BASIC programmiert haben, schlagen wir vor, daß Sie sich hierfür ein separates Lehrbuch zulegen oder an einem BASIC-Kurs teilnehmen, ehe Sie diese Kapitel durcharbeiten. Dieses Handbuch ist nicht als Lehrbuch für das Programmieren konzipiert.

Der übrige Teil des Handbuchs besteht aus:

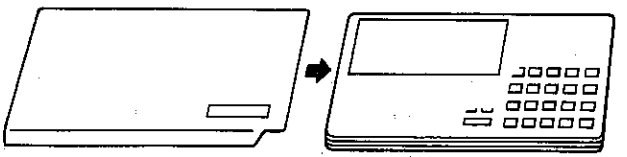
- Kapitel 7 — Grundlegende Informationen über den Gebrauch der wahlweisen Peripherigeräte.
- Kapitel 8 — Erklärungen zum Gebrauch der RAM-Karten.
- Kapitel 10 — Ein Leitfaden, der Ihnen helfen soll, Probleme der Bedienung und Programmierung zu lösen.
- Kapitel 11 — Pflege und Wartung Ihres neuen Computers.

Ein ausführlicher Anhang am Ende dieses Handbuches stellt Ihnen nützliche Tabellen, und einige weitere Hinweise zum Gebrauch und zur Funktionsweise des PC-1360 zur Verfügung.

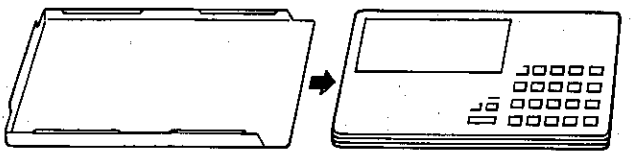
Entfernen und Einsetzen der Abdeckung



Wenn Sie den Computer gebrauchen, schieben Sie die Abdeckung heraus, kehren sie um und schieben sie wieder ein.



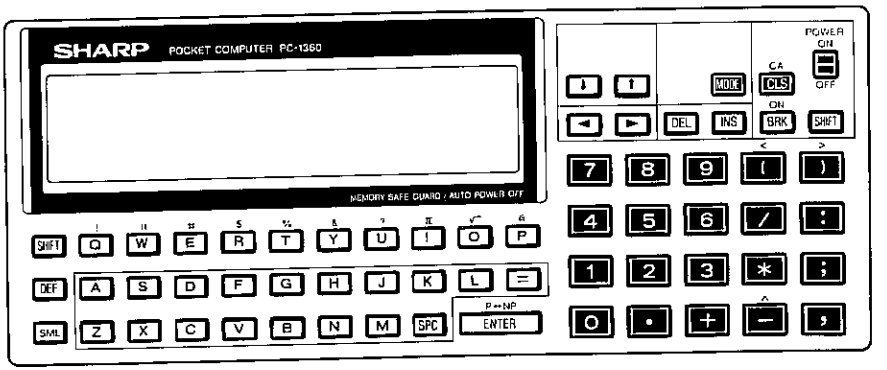
Verwenden Sie die Abdeckung, um den Computer zu schützen, wenn er nicht gebraucht wird.



2 EINFÜHRUNG IN DEN PC-1360

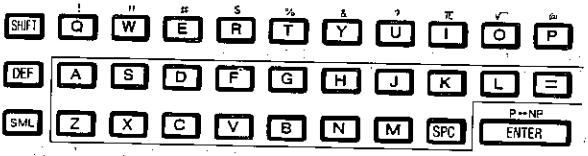
Das System SHARP PC-1360 besteht aus:

- 62-Zeichen-Tastatur.
- 4-zeilige 24-Zeichen-Anzeige oder Grafik-Display mit 150×32 Punkten.
- 8-Bit CMOS Prozessor.
- 136KB ROM.
- Erweiterter RAM-Speicher auf RAM-Karte (Standard 8KB).
- Eingänge für Druckerinterface und serielles E/A-Interface



Um Sie mit der Anordnung und den Funktionen von Teilen der Tastatur des PC-1360 vertraut zu machen, wollen wir nun die einzelnen Bestandteile der Tastatur genauer betrachten. Für den Anfang sollten Sie zunächst die Position der einzelnen Tasten feststellen und die dazugehörigen Beschreibungen lesen. In Kapitel 3 werden wir anfangen, mit Ihrem neuen Gerät zu arbeiten.

BESCHREIBUNG DER TASTATUR



A ~ **Z** Der Sharp PC-1360 hat die gleiche 26-Buchstaben-Tastatur, die man auf einer normalen Schreibmaschine findet. Normalerweise werden Großbuchstaben angezeigt (im Gegensatz zu den meisten Schreibmaschinen), das ist jedoch praktisch, da der PC-1360 nur Anweisungen und Kommandos in Großbuchstaben versteht. Zur Eingabe von Kleinbuchstaben, muß die Taste **SML** gedrückt werden, um den PC-1360 in den KLEINBUCHSTABEN-Modus zu setzen. (Siehe nächste Seite.)

= Gleichheitszeichen. Auf dem PC-1360 wird diese Taste nicht benutzt, um das Ende einer Rechenoperation anzuzeigen; dieses Symbol hat eine spezielle Funktion in der BASIC-Programmierung. SPC Leertaste. Nach Betätigen dieser Taste rückt der Cursor um einen Schritt nach rechts, wobei ein Leerfeld hinterlassen wird. Betätigt man **SPC**, während der Cursor auf einem Buchstaben steht, wird dieser Buchstabe gelöscht.

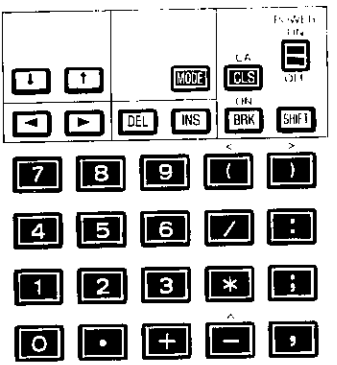
P←NP
ENTER **ENTER**-Taste. Wenn man diese Taste betätigt, wird, was immer vorher eingegeben wurde, in den Speicher des Computers befördert. Die Funktion ähnelt der Wagenrücklauftaste auf einer Schreibmaschine. Man muß **ENTER** drücken, bevor der PC-1360 auf alphanumerische Eingaben über die Tastatur reagiert.

DEF **DEF**-Taste. Diese spezielle Taste wird zum Ablauf von BASIC-Programmen benutzt.

SHIFT Die **SHIFT** hat mehrere wichtige Funktionen. Erstens, wenn Sie diese Taste vor einer anderen Taste betätigen, auf der eine zweite Funktion oben abgedruckt ist, wird diese zweite Funktion angezeigt (d.h. wenn Sie die **SHIFT**-Taste vor der **CLS**-Taste drücken, wird die Clear All-Funktion (CA) aktiviert). Zweitens, wenn Sie die **SHIFT**-Taste vor der Taste **MODE** betätigen, wird der RESERVE-Modus aktiviert. Normalerweise wird mit der **MODE**-Taste der RUN- oder PROGRAM-Modus aufgerufen. Durch die Benutzung der Taste **SHIFT**, ergibt sich nun eine dritte Möglichkeit. (Lesen Sie dazu auch die entsprechenden MODUS-Abschnitte in diesem Kapitel!) Drittens, kann die im RESERVE-Modus gespeicherte Zeichenfolge durch Betätigen der **SHIFT**-Taste und der Taste, unter welcher die Zeichenfolge gespeichert ist, abgerufen werden.

SML Die Taste **SML** wird für die Ausgabe von Kleinbuchstaben benutzt. Wenn man diese Taste einmal drückt, wird der PC-1360 in den KLEINBUCHSTABEN-Modus gesetzt ("SML" wird angezeigt). Wenn man sie erneut drückt, wird der PC-1360 in den GROSSBUCHSTABEN-Modus zurückgesetzt ("SML" verschwindet).

! " # $ % & ? π √ @ Diese Symbole befinden sich über der oberen Reihe der Alpha-Tastatur. Betätigt man **SHIFT** und dann den Buchstaben unter dem entsprechenden Zeichen, erscheinen diese Symbole in der Anzeige.



POWER ON OFF

Benutzen Sie diesen Schiebeschalter, um den PC-1360 ON/OFF ein- bzw. auszuschalten.

SHIFT **SHIFT**-Taste. Siehe Seite 5.

CA
CLS Die rote **CLS**-Taste erlaubt es Ihnen, den gesamten Bildschirminhalt zu löschen. Wenn Sie **SHIFT** vor **CLS** drücken, aktivieren Sie die zweite Clear All-Funktion. Dadurch wird der Computer initialisiert, wobei die Anzeige und der Matrixblock gelöscht werden.

BEACHTEN SIE:
Die Clear All-Funktion löscht nicht alle Programme, Reservespeicher und Variablen. Um alle Programme zu löschen, geben Sie **NEW** ein und drücken **ENTER**, während Sie sich im PROGRAM-Modus befinden. Um alle Reservespeicher zu löschen, geben Sie **NEW** ein und drücken **ENTER**, während Sie sich im RESERVE-Modus befinden. Um alle Variablen auf Null zu setzen, geben Sie **CLEAR** ein und drücken **ENTER**.

MODE Wenn Sie den PC-1360 einschalten, achten Sie darauf, welchen Modus Sie gewählt haben (RUN, PROGRAM oder RESERVE). Drücken Sie die Taste **MODE**, um den jeweiligen Modus von RUN zu PROGRAM oder von PROGRAM zu RUN zu ändern. Drücken Sie **SHIFT** und dann **MODE**, um den RESERVE-Modus auszuwählen. Um den RESERVE-Modus zu verlassen, drücken Sie **MODE**. Alle manuellen Berechnungen (wenn Sie den Computer als Rechner benutzen) werden im RUN-Modus ausgeführt und alle Programme laufen im RUN-Modus. alle Programme werden im PROGRAM-Modus geschrieben und ausgegeben. Der RESERVE-Modus wird benutzt, um häufig benutzte Funktionen auf einzelnen Tasten zu speichern und bei gespeicherten Menüs wird er dazu benutzt, diese Funktionen schnell zu identifizieren.

ON
BRK BREAK-Taste. Die **BRK**-Taste wird zur Unterbrechung des Programmlaufs benutzt. Nach Abschalten des Rechners durch die Automatik wird er mit dieser Taste wieder eingeschaltet.

INS Einfügungstaste. Wenn man diese Taste drückt, setzt sie einen Leerschritt unmittelbar vor das Zeichen, das der Cursor jeweils anzeigt. Sie können dann hier neue Zeichen durch Betätigen irgendeiner Taste einfügen.

DEL Löschtaste. Diese Taste ermöglicht das Löschen jedes Zeichens, das vom Cursor gerade angezeigt wird.

↓ Pfeil-nach-unten-Taste. Diese Taste bewirkt, daß die nachstehende Zeile angezeigt wird.

↑ Pfeil-nach-oben-Taste. Diese Taste bewirkt, daß die vorherige Zeile ins Display gerufen wird.

◄ Rückschritt-Taste. Mit Hilfe dieser Taste können Sie den Cursor nach links bewegen, ohne dabei vorher eingegebene Zeichen zu löschen.

► Vorwärts-Taste. Mit Hilfe dieser Taste können Sie den Cursor nach rechts bewegen, ohne dabei vorher eingegebene Zeichen zu löschen.

0 ~ **9** **.** Zifferntasten. Die Anordnung dieser Tasten ähnelt der auf einem gewöhnlichen Taschenrechner.

/ Divisionstaste. Drücken Sie diese Taste, um das Divisionszeichen in Rechenoperationen einzuschliessen.

***** Multiplikationstaste. Drücken Sie diese Taste, um das Multiplikationszeichen in Rechenoperationen einzuschließen.

^
− Subtraktionstaste. Drücken Sie diese Taste, um das Subtraktionszeichen in Rechenoperationen einzuschließen. Wenn Sie **SHIFT** und dann diese Taste drücken, fungiert sie als Exponentiationszeichen, das anzeigt, daß eine Zahl potenziert werden soll.

+ Additionstaste. Drücken Sie diese Taste, um das Additionszeichen in Rechenoperationen einzuschliessen.

<

(Klammerzeichen links. Wenn Sie SHIFT und diese Taste drücken, wird die Relation "kleiner als" hergestellt.

>

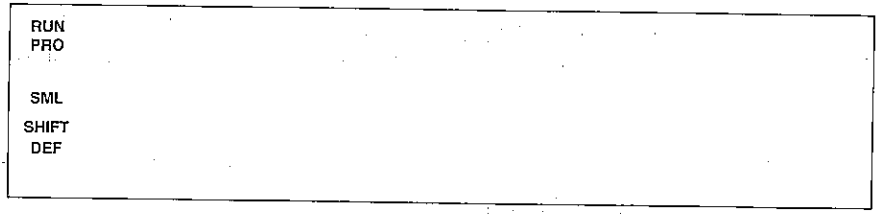
) Klammerzeichen rechts. Wenn Sie SHIFT und diese Taste drücken, wird die Relation "größer als" hergestellt.

: Doppelpunkttaste.

; Semikolontaste.

, Kommataste.

BESCHREIBUNG DES ZEILENDISPLAYS



Der PC-1360 hat eine programmierbare 150×32 Punkt-Flüssigkristallanzeige. Das Zeichendisplay besteht aus 4 Zeilen von je 24 Zeichen. Jedes Zeichen braucht zur seiner Darstellung 5×7 Punkte.

Bei graphischen Darstellungen kann das gesamte Display als 150×32 Punkt-Matrix benutzt werden. Einzelne Punkte können innerhalb der 150 Spalten so besetzt werden, daß man mit ihrer Hilfe Graphiken, beliebige Figuren oder spezielle Symbole erzeugen kann.

Das Anzeigefeld besteht aus:

Bereitschaftssymbol. Dieses Symbol erscheint, wenn der Computer bereit ist, eine Eingabe aufzunehmen. Sobald Sie mit der Eingabe beginnen, verschwindet das Bereitschaftssymbol und wird durch den Cursor ersetzt.

_ Cursor. Dieses Symbol (Unterstrich) markiert die Stelle des nächsten einzugebenden Zeichens. Wenn man mit der Eingabe von Daten beginnt, ersetzt der Cursor das Bereitschaftssymbol. Als Markierungssymbol wird der Cursor auch im Zusammenhang mit den INSert- und DELete-Funktionen benutzt.

RUN RUN-Symbol. Dieses Symbol zeigt den RUN-Modus für den PC-1360 an.

PRO PROgramm-Symbol. Dieses Symbol zeigt den Programmierungsmodus für den PC-1360 an.

Beachten Sie:
Wenn im Display weder RUN noch PRO erscheinen, befindet sich der PC-1360 im Reservemodus (RSV).

SML Anzeige für Kleinschriftmodus. Wenn man **SML** drückt, erscheint "SML" im Display. Wenn "SML" im Display erscheint, werden bei Betätigung der Alphatastatur Kleinbuchstaben angezeigt. Wenn man **SML** drückt, während das Symbol "SML" angezeigt wird, verschwindet das Symbol und der Computer kehrt in den Großbuchstabenmodus zurück.

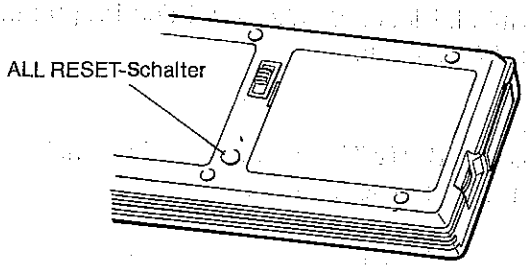
SHIFT Anzeige der Umschalttaste. Diese Anzeige leuchtet auf, wenn die **SHIFT**-Taste betätigt wurde. Bitte beachten Sie, daß die **SHIFT**-Taste wieder losgelassen werden muß, bevor eine andere Taste betätigt wird.

DEF Anzeige für Define-Modus. Dieses Symbol leuchtet auf, wenn die **DEF**-Taste betätigt wird.

Bei Eingaben, die länger sind als 4 Zeilen.
Das Display des PC-1360 besteht aus 4 Zeilen (je 24 Zeichen pro Zeile). Alle Eingaben oder Rechenergebnisse werden jeweils vom oberen Rand aus angezeigt. Falls die eingegebenen Zeichen 4 Zeilen überschreiten, wird der Bildschirminhalt einfach jeweils um 1 Zeile nach oben verschoben (die erste zeile der Anzeige rollt nach oben und ist somit auf dem Display nicht mehr zu sehen.)

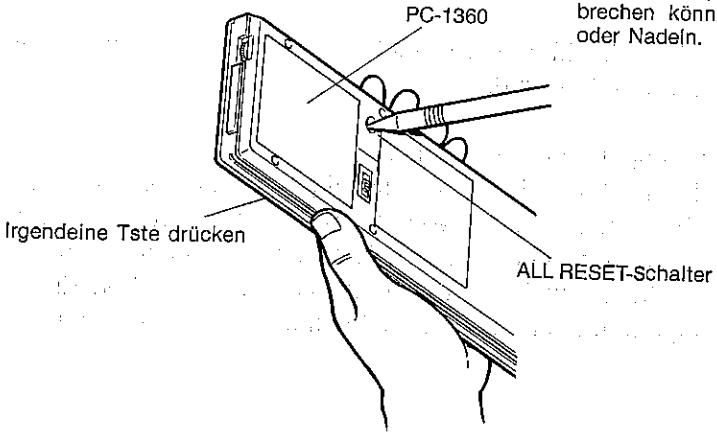
ALL-RESET-Schalter

ALL RESET: Rücksetzschalter. Er wird gebraucht, um den Computer zu initialisieren, wenn CLEAR (**CLS**) oder CA nicht austreichen, um das Problem zu lösen.
Bei Rücksetzvorgängen sollte der Computer immer eingeschaltet sein.

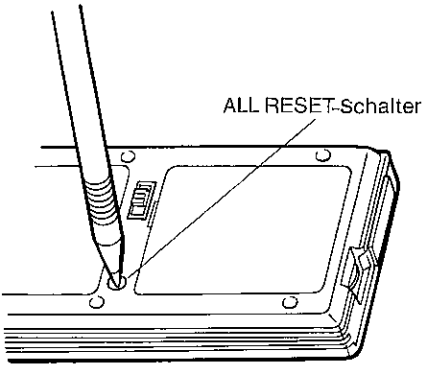


Beachten Sie:
Um den PC-1360 zu initialisieren, drücken Sie irgendeine Taste auf der Tastatur, halten diese fest und drücken gleichzeitig den ALL RESET-Schalter auf der Rückseite des Gerätes. Die Programme, Variablen und der Reservespeicher, einschließlich des RAM-kartenspeichers bleiben dabei erhalten.

Zum Drücken des ALL RESET-Schalters benutzen Sie bitte einen spitzen Gegenstand wie z.B. einen Kugelschreiber. Vermeiden Sie Gegenstände, deren Spitzen abbrechen könnten wie Bleistifte oder Nadeln.

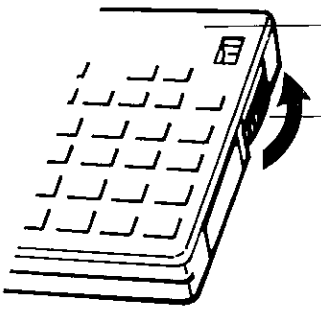


Sollte das Gerät nach dieser Operation immer noch nicht richtig funktionieren, versuchen Sie nur den ALL RESET-Schalter zu betätigen. Beachten Sie bitte, daß dabei allerdings alle Programme, Variablen und der Reservespeicher gelöscht werden, einschließlich des Inhaltes der RAM-Karte, falls vorhanden.



Wenn das Gerät anschließend noch nicht normal arbeitet, enfernen Sie bitte die Lithium-Batterien. Nach etwa 10 Sekunden setzen Sie die Batterien wieder ein und drücken RESET.

KONTRASTREGULIERUNG



Durch Drehen des Knopfes in Pfeilrichtung wird der Kontrast der Anzeige verstärkt und in entgegengesetzter Richtung verringert. Bitte stellen Sie ihn so ein, daß die Anzeige gut sichtbar ist.

BATTERIEWECHSEL

Der PC-1360 arbeitet ausschließlich mit Lithium-Batterien. Wenn er an die Zusatzeinrichtung CE-126P angeschlossen ist, kann der PC-1360 auch über diese versorgt werden, sofern deren momentane Betriebsspannung höher ist. Dies verringert den Stromverbrauch der Lithium-Zellen.

Auswechseln der Batterien:

- Wechseln Sie grundsätzlich beide Batterien, verwenden Sie niemals eine neue Batterie gemeinsam mit einer gebrauchten.
- Benutzen Sie nur Lithium-Zellen des Typs CR-2032.

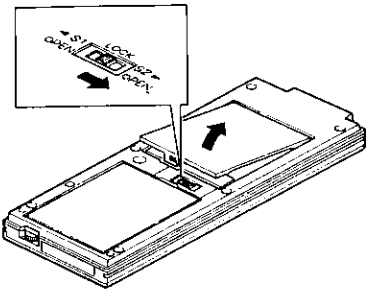
EINSETZEN DER BATTERIEN

Wenn die Anzeige schlecht zu erkennen ist, auch nach der Regulierung mit dem Kontrastknopf, müssen die Batterien ausgewechselt werden. Lassen Sie niemals verbrauchte Batterien im Computer, da diese auslaufen und das Gerät beschädigen können.

VORSICHT: Bewahren Sie die Batterien außerhalb der Reichweite von Kindern auf!

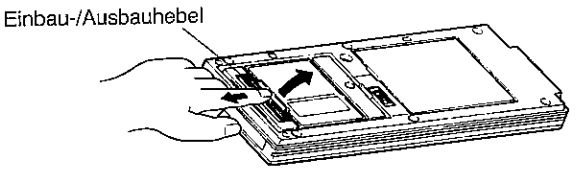
1.
Schalten Sie den Computer ab.

2.
Drehen Sie das Computer herum und schieben die Arretierung in Richtung des 2. Schlitzes. Nehmen Sie die Abdeckung des 2. Schlitzes ab. Wenn sich im 2. Schlitz keing Karte befindet, setzen Sie den Deckel wieder ein und verriegeln ihn. Wenn eine RAM-Karten im Schlitz ist, nehmen Sie sie heraus wie im Schritt 4 beschrieben.

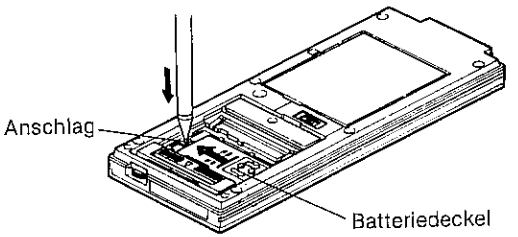


3.
Schieben Sie die Arretierung in die entgegengesetzte Richtung und nehmen die Abdeckung des 1. Schlitzes ab.

4.
Um die eingesetzte Karte her- auszunehmen, drücken Sie die Karte in die Pfeilrichtung.

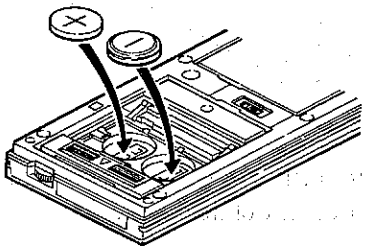


5.
Drücken Sie leicht auf den Anschlag und schieben Sie den Batteriedeckel in die angegebene Richtung, um ihn zu entfernen.



6.

Nehmen Sie die verbrauchten Batterien heraus und wechseln sie gege
zwei neue Lithium-Batterien aus. Achten Sie darauf, daß die neuen Batte
rien mit einem trockenen Tuch gereinigt werden und mit der korrekten Po
larität eingelegt werden.



7.

Setzen Sie den Batteriedeckel in der gleichen Weise wie im Schritt 5 be
schrieben wieder ein.

8.

Stellen Sie die Arretierung auf die Verriegelungsseite für den 1. Schlitz. In
stallieren Sie die RAM-Karten noch nicht.
Legen Sie die Abdeckung des 1. Schlitzes wieder ein und verriegeln ihn.
Die RAM-Karten sollten noch nicht eingelegt werden. Wenn Sie jetzt eine
RAM-Karte einlegen, wird ihr Speicherinhalt durch die Betätigung des All
Reset-Schalters im Schritt 9 gelöscht.
Die Abdeckung auf der Rückseite des Geräts muß korrekt verriegelt sein,
damit der Computer funktionieren kann. Sollten Sie dies vergessen, schlie-
ßen Sie die Abdeckung auf der Rückseite, schalten den Computer ab und
dann wieder ein.

9.

Schalten Sie den Computer ein und drücken den All Reset-Schalter auf der
Unterseite des Geräts.

10.

Vergewissern Sie sich, daß das Sternzeichen in der rechten oberen Ecke
der Anzeige erscheint. Wenn es nicht erscheint, entfernen Sie die Batte-
rien, reinigen sie und setzen sie wieder ein.

```
                    *
```

11.
Mit abgeschaltetem Computer legen Sie die RAM-Karte in den 1. Schlitz.

Falls der PC-1360 nicht richtig funktioniert, kann der Grund folgender sein:

1/ Der Kontrast des Bildschirms ist schlecht; stellen Sie ihn mit dem Kontrastknopf ein.

2/ Wenn die Anzeige immer noch das Sternzeichen in der rechten oberen Ecke angibt, ist die RAM-Karte nicht im 1. Schlitz.

3/ Die Batterien im PC-1360 sind langsam verbraucht.

4/ Die Speichergröße ist nicht richtig. Die RAM-Karte wurde schlecht in den 1. Schlitz eingelegt.

Wenn der PC-1360 nicht richtig funktioniert, setzen Sie das Gerät mit Hilfe des All Reset-Schalters zurück. Wenn der All Reset-Schalter allein gedrückt wird, ohne eine andere Taste zu betätigen, erscheint die folgende Anzeige:

```
MEM$ = "C"
RAM CARD S1 CLEAR O.K. ?
```

Wird darauf die **Y**-Taste gedrückt, so wird die RAM-Karte gelöscht und alle Programme und Daten gehen verloren. Falls Probleme auftreten sollten, drücken Sie irgendeine Taste außer der **Y**-Taste und drücken gleichzeitig den All Reset-Schalter. Die folgende Anzeige erscheint:

```
RUN MODE
>
```

VOR DEM GEBRAUCH DES PC-1360

Vergewissern Sie sich, daß die Batterien und die 8KB RAM-Karte (CE-212M) richtig eingelegt sind. Schalten Sie den Hauptschalter oben rechts auf dem Gerät ein. Da die RAM-Karte im 1. Schlitz neu ist und noch nicht initialisiert wurde, erscheint folgende Anzeige:

```
MEM$ = "C"
RAM CARD S1 CLEAR O.K. ?
```

drücken Sie die **Y**-Taste, um den Computer zu initialisieren, und folgende Anzeige zu erhalten:

```
RUN MODE
>
```

Um sich zu vergewissern, daß der Computer richtig initialisiert wurde, geben Sie folgendes ein:

MEM ENTER

Wenn alles OK ist, sehen Sie folgende Anzeige:

```
RUN MODE
MEM
                    6558.
```

Bei RAM-Karten anderer Größe wird diese Darstellung anders sein, für die mit dem Computer mitgelieferte Karte CE-212M sollte sie aber immer 6558 sein. Nachfolgend werden Größen für andere Karten aufgelistet.

Typ der RAM-Karte	Verfügbare Bytes
CE-210M (2KB)	414.
CE-211M (4KB)	2462.
CE-212M (8KB)	**6558.**
CE-2H16M (16KB)	14750.
CE-2H32M (32KB)	31134.

Wenn etwas nicht stimmt, zeigt die Anzeige folgendes an:

In diesem Fall drücken Sie den All Reset-Schalter auf der Rückseite des Computers und versuchen, die Initialisierung wieder von vorne anzufangen.

EINSCHALTEN

Schalten Sie den Hauptschalter oben rechts auf dem Gerät ein. Falls die RAM-Karte nicht initialisiert ist, erscheint die folgende Anzeige:

```
RUN MODE
>
```

Wenn seit der letzten Verwendung des Computers eine neue RAM-Karte eingeschoben wurde, wird wieder die Initialisationsanzeige erscheinen. Gehen Sie wie im Vorangehenden Abschnitt beschrieben vor, ehe Sie den PC-1360 benutsen.

3 DER GEBRAUCH DES PC-1360 ALS RECHNER

Nachdem Sie nun vertraut sind mit den Bestandteilen des SHARP PC-1360 und deren Anordnung, wollen wir damit beginnen, die faszinierenden Fähigkeiten Ihres neuen Computers zu entdecken.

Weil der PC-1360 über alle gebräuchlichen mathematischen Funktionen verfügt und darüber hinaus in BASIC programmierbar ist (nützlich für komplexere Rechenoperationen), wird er häufig als "schlauer" Rechner bezeichnet. Das macht natürlich aus Ihnen einen "schlauen" Benutzer!

(Bevor Sie nun mit dem PC-1360 arbeiten, vergewissern Sie sich, daß die Batterien korrekt eingesetzt sind.)

Einschalten des Computers

Um den PC-1360 einzuschalten, schieben Sie den Schiebeschalter nach oben. Für den Gebrauch als Taschenrechner, muß sich der PC-1360 im RUN-Modus befinden. Benutzen Sie dazu die **MODE**-Taste, bis "RUN" an der linken Seite des Anzeigefeldes erscheint. Wenn das Gerät eingeschaltet ist, erscheint das Bereitschaftssymbol (>) in der Anzeige.

Ausschalten des Computers

Um den PC-1360 auszuschalten, bringen Sie den Schiebeschalter in die Position OFF.

Wenn Sie das Gerät auf OFF schalten, löschen Sie das Zeilendisplay. Der PC-1360 merkt sich jedoch alle Programme und Reservetasten-Einstellungen, die im Einsatz waren, als der Computer ausgeschaltet wurde. All diese Einstellungen stehen wieder zur Verfügung, sobald das Gerät wieder eingeschaltet wird.
Wenn der CLOAD-Befehl ausgeführt wird, unterbrechen Sie die Ausführung, indem Sie die **BRK**-Taste drücken und den Schiebeschalter in die Position OFF bewegen.

Automatische Abschaltung

Um die Batterien vor unnützer Beanspruchung zu schützen, schaltet sich der PC-1360 automatisch ab, wenn 11 Minuten lang keine Taste betätigt wurde. (Achtung: Die automatische Abschaltung wird ausgesetzt, solange der PC-1360 ein Programm abarbeitet.)

Wenn Sie den Computer nach automatischer Abschaltung wieder starten wollen, tun Sie das mit der ON **BRK** -Taste. Alle Einstellungen bleiben genauso wie sie waren, als der Computer automatisch abgeschaltet wurde.

Einige hilfreiche Hinweise

Bis Sie sich an Ihr neues Gerät gewöhnt haben, werden Sie sicherlich Fehler bei der Dateneingabe machen. Wir werden später einige einfache Korrekturmöglichkeiten kennenlernen. Für's erste, wenn Sie eine Fehlermeldung erhalten, drücken Sie de rote CLEAR-Taste (**CLS**) und wiederholen die Eingabe. Wenn der Computer "aufhängt" - d.h. überhaupt nicht mehr reagiert — betätigen Sie den ALL RESET-Schalter (siehe Kapitel 2).

Das Bereitschaftssymbol (>) sagt Ihnen, daß der PC-1360 eine Eingabe erwartet. Wenn Sie Daten eingeben, verschwindet das Bereitschaftssymbol und der Cursor bewegt sich nach rechts und zeigt dabei die nächstverfügbare Stelle im Zeilendisplay an.

Mit den Pfeilen nach rechts **▶** und links **◀** bewegen Sie den Cursor innerhalb einer Zeile.

Wenn Sie **ENTER** drücken, informieren Sie den PC-1360 darüber, daß Sie die Dateneingabe beendet haben und Sie fordern den Computer auf, die gewünschten Operationen durchzuführen. SIE MÜSSEN **ENTER** AM ENDE JEDER EINGABEZEILE BETÄTIGEN; ANDERNFALLS WERDEN IHRE RECHNUNGEN VOM COMPUTER NICHT AUSGEFÜHRT WERDEN.

Wenn Sie numerische Berechnungen durchführen, erscheint die Eingabe links im Display; das Ergebnis steht rechts im Display.

Benutzen Sie die SHIFT-Taste in Verbindung mit einer anderen Taste (um z.B. die Quadratwurzel zu ziehen), drücken Sie SHIFT, lassen SHIFT wieder los und drücken dann die andere Taste. Der SHIFT-Modus bezieht sich lediglich auf die nächste gedrückte Taste.

Benutzen Sie keine $-Zeichen oder Kommata, wenn Sie mit dem PC-1360 ohne Programmunterstützung rechnen. Diese Zeichen haben eine besondere Bedeutung in der Programmiersprache BASIC. In diesem Handbuch benutzen wir die Ø, um die Null zu kennzeichnen, so daß zwischen der Ziffer (Ø) und dem Buchstaben (O) unterschieden werden kann.

Um Ihnen die korrekte Dateneingabe am Anfang zu erleichtern, werden wir in den Rechenbeispielen jede nötige Tastenbedienung zeigen. Wenn die SHIFT-Funktion angesprochen wird, werden wir das gewünschte Zeichen im folgenden Tastensymbol abbilden. So entsteht beispielsweise durch Betätigen von SHIFT und $\frac{!}{a}$ das !-Zeichen. Eine solche Tastenbedienung wird mit SHIFT ! dargestellt.

Denken Sie daran, nach jeder Rechenoperation die CLS-Taste zu betätigen (es sei denn, Sie möchten Kettenrechnungen durchführen). CLEAR löscht die Anzeige und beseitigt eventuelle Blockaden durch Fehler. Speicherinhalte werden nicht gelöscht.

Einfache Rechenoperationen

Der PC-1360 rechnet auf 1 Stelle genau. Schalten Sie Ihren Computer ein und bringen Sie ihn in den RUN-Modus. Nun versuchen Sie, die folgenden einfachen Rechenbeispiele nachzuvollziehen.

Eingabe	Anzeige
5 0 + 5 0 ENTER	5Ø+5Ø 1ØØ.
1 0 0 − 5 0 ENTER	5Ø+5Ø 1ØØ. 1ØØ−5Ø 5Ø.

6 0 * 1 0 ENTER

```
100-50
                                50.
60*10
                               600.
```

Das Display des PC-1360 besteht aus 4 Zeilen (je 24 Zeichen pro Zeile). Alle Eingaben oder Rechenergebnisse werden jeweils von der obersten Zeile an angegeben. Wenn die Daten die Kapazität von 4 Zeilen überschreiten, wird der Anzeigeinhalt jeweils um 1 Zeile nach oben gerollt (die erste angezeigte Zeile rollt nach oben und verschwindet dann aus dem Display).

Eingabe

3 0 0 / 5 ENTER

Anzeige

```
60*10
                               600.
300/5
                                60.
```

1 0 SHIFT ^ 2 ENTER

```
300/5
                                60.
10^2
                               100.
```

2 * SHIFT π ENTER

```
10^2
                               100.
2*π
                          6.283185307
```

SHIFT √ 6 4 ENTER

```
2*π
                          6.283185307
√ 64
                                 8.
```

4 E 3 ENTER

```
√ 64
                    8.
4E3
                  4ØØØ.
```

Wiederabruf von Eingaben

Auch nachdem der PC-1360 die Ergebnisse Ihrer Rechenoperation angezeigt hat, können Sie Ihre vorherige Eingabe aufbereiten lassen. Dazu benutzen Sie die Rechts-► und Links-◄ Pfeile.

Der Links-◄ Pfeil wird benutzt, um den Cursor an die Stelle hinter dem letzten Zeichen zu positionieren.

Der Rechts-► Pfeil wird benutzt, um den Cursor auf das erste Zeichen zu setzen.

Sie erinnern sich, daß der Rechts- und Links-Pfeil auch benutzt werden, um den Cursor innerhalb der Zeile zu bewegen. Die Pfeile sind sehr hilfreich beim Kontrollieren (oder Verändern) von Eingaben, ohne dabei den gesamten Ausdruck noch einmal eingeben zu müssen.

Beim Durchlesen der folgenden Beispiele sollten Sie versuchen, diese gleichzeitig auf dem Computer auszuführen, um sich mit dem Gebrauch der Tasten vertraut zu machen.

Als Personalchef in einem großen Marketing-Unternehmen sind Sie verantwortlich für die Planung der jährlichen Verkaufstagung. Sie erwarten zu diesem dreitätigen Treffen 300 Personen. Für einen Teil dieser Zeit wird man sich in kleine Arbeitsgruppen aufteilen. Sie sind der Ansicht, daß Gruppen zu je 6 Personen eine gute Größe wären. Wieviele Gruppen müssen Sie einplanen?

Eingabe

3 0 0 / 6 ENTER

Anzeige

```
3ØØ/6
                   5Ø.
```

Dann fällt Ihnen ein, daß Gruppen mit einer ungeraden Teilnehmerzahl möglicherweise effektiver arbeiten. Rufen Sie Ihre letzte Eingabe mit Hilfe des ◄-Pfeils wieder auf.

Eingabe	Anzeige
◄	300/6_

Um die neue Anzahl der Gruppen zu berechnen, müssen Sie die 6 durch eine ungerade Zahl ersetzen. Fünf Gruppenmitglieder scheinen Ihnen sinnvoller als sieben. Da Sie daran gedacht haben, den ◄-Pfeil zu benutzen, steht der Cursor nun am Ende der Anzeige. Benutzen Sie noch einmal den ◄-Pfeil, um den Cursor eine Stelle nach links zu bewegen.

Eingabe	Anzeige
◄	300/6

Beachten Sie, daß nach dieser Bewegung der Cursor sich in ein blinkendes Viereck verwandelt. Immer, wenn Sie den Cursor auf ein schon bestehendes Zeichen setzen, wird er in dieser Form dargestellt.

Geben Sie nun eine 5 anstelle der 6 ein. Achtung beim Ersetzen von Zeichen: Wenn Sie ein neues Zeichen über bereits bestehendes setzen, ist das ursprüngliche endgültig gelöscht! Sie können keinen Ausdruck wieder abrufen, der überschrieben wurde.

Eingabe	Anzeige
5 ENTER	300/5 60.

Sechzig scheint Ihnen eine vernünftige Anzahl von Gruppen zu sein, so beschließen Sie, daß jede Kleingruppe aus 5 Teilnehmern bestehen wird.

Erinnern Sie sich, daß es sinnvoll ist, die letzte Eingabe zu kontrollieren, besonders, wenn Ihre Ergebnisse Ihnen zweifelhaft erscheinen. Nehmen Sie z.B. an, Sie hätten folgende Rechnung durchgeführt:

Eingabe	Anzeige
CLS 3 0 / 5 ENTER	30/5 6.

Selbst ein müder, überarbeiteter Manager, wie Sie es sind, bemerkt, daß 6 kein glaubwürdiges Ergebnis ist, wenn man mit Hunderten von Personen operiert! Rufen Sie Ihre Eingabe mit Hilfe des ►-Pfeils noch einmal ins Zeilendisplay.

Eingabe	Anzeige
►	30/5

Weil Sie den ►-Pfeil benutzt haben, steht der blinkende Cursor jetzt auf dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, möchten Sie jetzt eine zusätzliche Null einfügen. Mit Hilfe des ►-Pfeils bewegen Sie den Cursor, bis er über der Null steht. Wenn Sie nun mit der INS-Taste eine Einfügung machen wollen, setzen Sie den blinkenden Cursor auf denjenigen Buchstaben, vor dem das neue Zeichen eingefügt werden soll.

Eingabe	Anzeige
►	30/5

Benutzen Sie die INS-Taste, um Platz für das benötigte Zeichen zu schaffen.

Eingabe	Anzeige
INS	3 0/5

Das Betätigen der INS-Taste bewegt alle Zeichen um einen Schritt nach rechts und fügt einen von Klammern eingeschlossenen Leerraum ein. Der blinkende Cursor steht nun auf diesem Leerraum und zeigt die Stelle der nächsten Eingabe an. Geben Sie die fehlende Null ein. Nachdem die Eingabe korrigiert ist, rufen Sie erneut Ihr Ergebnis ab.

Eingabe	Anzeige
0 ENTER	300/5 60.

Stellen Sie sich auf der anderen Seite vor, Sie hätten die folgende Eingabe gemacht:

Eingabe	Anzeige
CLS 3 0 0 0 / 5 ENTER	3000/5 600.

Das Ergebnis scheint viel zu groß. Wenn Sie nur 300 Personen zu Ihrem Treffen erwarten, wie könnte es dann 600 Arbeitsgruppen geben? Rufen Sie Ihre Eingabe mit Hilfe des ► -Pfeils ab.

Eingabe	Anzeige
►	3000/5

Der blinkende Cursor steht nun über dem ersten Zeichen der Anzeige. Um die Eingabe zu korrigieren, müssen Sie eine der Nullen löschen. Mit Hilfe des ► -Pfeils bewegen Sie den Cursor auf die erste (oder eine andere) Null. Wenn Sie ein Zeichen löschen wollen, setzen Sie den Cursor auf das zu löschende Zeichen.

Eingabe	Anzeige
►	3000/5

Nun betätigen Sie die DEL-Taste, um eine der Nullen zu beseitigen.

Eingabe	Anzeige
DEL	3Ø0/5

Durch Betätigen der DEL-Taste rutschen alle Zeichen um einen Schritt nach links. Das Zeichen, auf dem der Cursor steht, wird gelöscht, ebenso der Freiraum, den dieses Zeichen beansprucht. Der blinkende Cursor bleibt an derselben Stelle stehen und zeigt die nächste Stelle für eine Eingabe an. Da Sie keine weiteren Veränderungen vorzunehmen haben, beenden Sie die Rechnung.

Eingabe	Anzeige
ENTER	3Ø0/5 6Ø.

(Achtung: Steht der Cursor auf einem Zeichen und Sie betätigen die SPC-Taste, wird das betreffende Zeichen gelöscht und durch einen Leerschritt ersetzt. Mit DEL löschen Sie das Zeichen und den von ihm eingenommenen Leerraum.)

Fehler

Der Abruf der vorherigen Eingabe ist notwendig, wenn Sie die gefürchtete ERROR-Meldung erhalten. Wir wollen uns vorstellen, daß Sie unabsichtlich folgendes in den PC-1360 eingegeben haben:

Eingabe	Anzeige
CLS 3 0 0 / / 5 ENTER	300//5 ERROR 1

Natürlich sind Sie überrascht, wenn diese Meldung erscheint! ERROR 1 ist einfach die Art des Computers, Ihnen mitzuteilen: "Ich weiß nicht, was ich hier tun soll."

Wenn Sie in diesem Fall den ◄-Pfeil betätigen, erscheint der blinkende Cursor an der Stelle, wo der Fehler liegt.

Eingabe	Anzeige
◄	300/5

Um diesen Fehler zu korrigieren, benutzen Sie die DEL-Taste.

Eingabe	Anzeige
DEL ENTER	300/5 60.

Wenn Sie beim Abruf Ihrer Eingabe aufgrund der Meldung ERROR 1 feststellen, daß Sie ein Zeichen vergessen haben, benutzen Sie die INS-Sequenz, um es zu korrigieren.
Wenn Sie den PC-1360 wie einen normalen Taschenrechner benutzen, werden Sie in der Hauptsache mit dem ERROR 1 (Syntaxfehler) konfrontiert werden. Eine komplette Aufstellung der Fehlermeldungen und ihrer Bedeutungen finden Sie in ANHANG A.

Verkettung von Rechenoperationen

Ihr PC-1360 erlaubt, daß Sie das Ergebnis einer Rechnung in der folgenden Operation weiterverwenden.
Zurück zu unserem Beispiel: Ein Teil Ihrer Verantwortung bei der Planung dieser Konferenz besteht darin, einen detaillierten Budget-Vorschlag auszuarbeiten. Sie wissen, daß Ihnen für jeden Teilnehmer DM 150,00 zur Verfügung stehen. Berechnen Sie Ihr Gesamtbudget.

Eingabe	Anzeige
CLS 3 0 0 * 1 5 0 ENTER	300*150 45000.

Von dieser Summe beabsichtigen Sie, 15% für die Abschlußfeier aufzuwenden. Wenn Sie verkettete Rechenoperationen durchführen, ist es nicht nötig, Ihr erstes Ergebnis noch einmal einzugeben, sofern Sie die CLS-Taste NICHT benutzen. Wie hoch ist Ihr Budget für die Feier?

Eingabe	Anzeige
[*] [.] [1] [5]	45000.*.15_

Beachten Sie, daß der Computer das Ergebnis Ihrer ersten Berechnung automatisch links im Zeilendisplay anzeigt und in die neue Rechnung miteinbezieht, wenn Sie die Befehle für die zweite Berechnung (*.15) eingeben. In verketteten Rechenoperationen muß die Eingabe mit einem Operationssymbol beginnen. Beenden Sie Ihre Eingabe wie immer mit [ENTER]:

ACHTUNG:
Die [%] [T]-Taste kann nicht in Rechenoperationen benutzt werden. Sie sollte ausschließlich als ein Zeichen benutzt werden.
Beispiel: 45000 [*] 15 [SHIFT] [%] [T] → ERROR 1

Eingabe	Anzeige
[ENTER]	6750.

Fahren Sie fort, Ihr Budget zu berechnen. Das Hotel verlangt für die Verpflegung DM 4000,--.

Eingabe	Anzeige
[-] [4] [0] [0] [0]	6750.-4000_
[ENTER]	2750.

Die Dekoration wird DM 1225,-- kosten.

Eingabe	Anzeige
[-] [1] [2] [2] [5] [ENTER]	1525.

Schließlich müssen Sie DM 2200,-- für den Sprecher und für das Rahmenprogramm einkalkulieren.

Eingabe	Anzeige
(-) 2 2 0 0 ENTER	-675.

Offensichtlich werden Sie entweder Ihre Pläne oder Ihre Kostenverteilung ändern müssen!

Negative Zahlen

Da die Abschlußfeier etwas ganz Besonderes sein soll, bechließen Sie, Ihre Pläne nicht zu ändern und das zusätzliche Geld zu investieren. Trotzdem fragen Sie sich, wieviel Prozent des Gesamtbudgets wohl von diesem Posten in Anspruch genommen werden. Zuerst ändern Sie das Vorzeichen Ihres letzten Ergebnisses.

Eingabe	Anzeige
* (-) 1	-675.*-1_

Eingabe	Anzeige
ENTER	675.

Nun addieren Sie dieses Ergebnis zu Ihrem ursprünglichen Budget für die Feier hinzu.

Eingabe	Anzeige
+ 6 7 5 0 ENTER	7425.

Teilen Sie Ihr neues Ergebnis durch 45000 und Sie erfahren, wieviel Prozent des Gesamtbudgets diese neue Zahl ausmacht.

Eingabe	Anzeige
I 4 5 0 0 0 ENTER	0.165

Nun gut, Sie beschließen, 16,5% für die Abschlußfeier aufzuwenden.

Komplexere Rechenoperationen und Klammerung

Bei der Durchführung der obigen Rechenbeispiele hätten Sie einige dieser Operationen in einem Schritt zusammenfassen können. So hätten Sie etwa die beiden folgenden Berechnungen in eine Zeile schreiben können:

675 + 6750/45000

Solche komplexeren Rechnungen müssen jedoch sehr sorgfältig eingegeben werden.

675 + 6750/45000 könnte verstanden werden als

$$\frac{675 + 67500}{45000} \qquad \text{oder} \qquad 675 + \frac{6750}{45000}$$

Bei der Durchführung komplexer mathematischer Operationen folgt der PC-1360 spezifischen Regeln der impliziten Klammerung und der Rangfolge von mathematischen Operatoren (siehe ANHANG D). Um sicherzustellen, daß Ihre Eingaben in der von Ihnen gewünschten Reihenfolge verarbeitet werden, sollten Sie Klammern verwenden.

(675 + 6750)/45000 oder 675 + (6750/45000)

Wie unterschiedlich sich die Plazierung von Klammern auf das Ergebnis auswirken kann, sehen Sie in den beiden folgenden Beispielen:

Eingabe

(6 7 5 + 6
7 5 0) / 4
5 0 0 0 ENTER

Anzeige

Ø.165

6 7 5 + (6
7 5 0 / 4 5 0 0
0) ENTER

675.15

Der Gebrauch von Variablen in Rechenoperationen

Der PC-1360 kann bis zu 26 einstellige Variablen unter den alphabetischen Zeichen A – Z speichern. Wenn Ihnen der Begriff der Variablen nicht vertraut ist: In Kapitel 4 finden Sie ausführliche Erklärungen. Die Variablenzuweisung ge4schieht über einen Zuweisungsbefehl:

A = 5
B = −2

Sie können auch den Wert einer Variablen (rechts) einer anderen (links) zuweisen.

C = A + 3
D = E

Eine Variable kann in jeder Rechenoperation anstelle einer Zahl benutzt werden.

Nachdem Sie nun Ihre Abschlußfeier geplant haben, fehlen Ihnen noch die übrigen Berechnungen für Ihre Verkaufstagung. Sie möchten den Rest Ihres Budgets ebenfalls prozentual aufteilen. Zuerst müssen Sie herausfinden, wieviel Geld Sie noch zur Verfügung haben. Weisen Sie eine Variable (R) dem verbleibenden Rest zu.

Eingabe	Anzeige
R = 4 5 0 0 0 − 7 4 2 5 ENTER	R = 45000 − 7425 37575.

Wenn Sie nun ENTER drücken, führt der PC-1360 die Berechnung durch und zeigt den neuen Wert von R an. Sie können den Wert jeder Variablen zur Anzeige bringen, indem Sie den Buchstaben eingeben, unter dem er gespeichert ist.

Eingabe	Anzeige
R ENTER	37575.

Nun können Sie Rechenoperationen mit Ihren Variablen durchführen. Der Wert von R wird sich nicht verändern, bis Sie ihm einen neuen Wert zuweisen.

Sie beabsichtigen, 60% des verbleibenden Geldes auf die Unterbringung zu verwenden.

Eingabe	Anzeige
R * . 6 0 ENTER	R*.60 22545.

Weiterhin wollen Sie 25% des verbleibenden Budgets für die Durchführung von Management-Trainings-Seminaren ausgeben.

Eingabe	Anzeige
R * . 2 5 ENTER	9393.75

Variablen behalten ihren zugewiesenen Wert auch wenn das Gerät ausgeschaltet wird oder sich automatisch abschaltet. Sie gehen nur verloren, wenn

- Man ihnen einen neuen Wert zuweist.
- Man CLEAR ENTER (nicht die CLS-Taste) eingibt.
- Man das Gerät mit dem ALL RESET-Schalter initialisiert.
- Die Batterien gewechselt werden.

Es gibt gewisse Grenzen der Variablenzuweisung sowie gewisse Programmabläufe, die Variablen verändern können. In Kapitel 4 finden Sie detaillierte Informationen zur Variablenzuweisung, in Kapitel 5 wird der Gebrauch von Variablen bei der Programmierung erörtert.

Mehrere Rechenschritte

Mit dem PC-1360 ist es nicht nur möglich, verschiedene Rechenschritte zusammenzufassen, man kann auch einzelne Rechnungen nacheinander durchführen, ohne zwischendurch ENTER betätigen zu müssen. Die einzelnen Rechenoperationen werden durch Kommata getrennt. In der Anzeige erscheint nur das Ergebnis der letzten Rechenoperation. (Beachten Sie bitte, daß die maximale Zeilenlänge, die Ihr Computer annimt, 80 Zeichen einschließlich ENTER beträgt.)

Sie fragen sich, wieviel Geld für die Unterbringung zur Verfügung stünde, wenn Sie, wie ursprünglich geplant, nur 15% für die Abschlußfeier aufwenden würden.

Eingabe	Anzeige
R =. 8 5 * 4 5 0 0 0 , R * . 6 0	R=.85*45000,R*.60_

Obwohl der Computer alle eingegebenen Rechenoperationen durchführt, zeigt er nur das letzte Ergebnis an.

Eingabe	Anzeige
ENTER	R=.85*45000,R*.60 22950.

Um herauszufinden, welchen Wert die Variable R in dieser Berechnung hatte, geben Sie R ein:

Eingabe	Anzeige
R ENTER	3825Ø.

Wissenschaftliche Notation

Wer mit sehr großen und sehr kleinen Zahlenwerten zu arbeiten hat, benutzt oft ein spezielles Format, die sogenannte exponentielle oder wissenschaftliche Notation. In der wissenschaftlichen Notation wird eine Zahl in zwei Teile aufgespalten.

Der erste Teil besteht aus einer Dezimalzahl zwischen 1 und 10. Der zweite Teil gibt die Größe der Zahl in Zehnerpotenzen an.

Wie Sie wissen, ist in einer gewöhnlichen Dezimalzahl die erste Stelle links vom Komma die Stelle der Einer, an zweiter Stelle folgen die Zehner, an dritter die Hunderter, dann die Tausender usw. Dies sind einfach ansteigende Potenzen von 10.

$10^0 = 1$, $10^1 = 10$, $10^2 = 100$, $10^3 = 1000$, usw.

Die wissenschaftliche Notation bricht also eine Dezimalzahl in zwei Teile auseinander. Ein Teil zeigt, um welche Ziffernfolge es sich handelt, der zweite, wieviele Stellen eine Zahl links oder rechts vom Komma aufweist. Z.B.:

1234 wird 1.234 mal 10^3 (3 Stellen nach rechts)
654321 wird 6.54321 mal 10^5 (5 Stellen nach rechts)
.000125 wird 1.25 mal 10^{-4} (4 Stellen nach links)

Wissenschaftliche Notation ist hilfreich, um Eingaben zu verkürzen. Um 1,0 mal 10^{87} einzugeben, müßte man eine 1 mit 87 Nullen schreiben! In wissenschaftlicher Notation aber würde diese Zahl folgendermaßen aussehen:

1.0×10^{87} oder 1.ØE87

Der PC-1360 benutzt die wissenschaftliche Notation immer dann, wenn Zahlen in dezimaler Schreibweise die Anzeigekapazität übersteigen würden. Dabei wird der Ausdruck "mal 10 hoch" durch den Großbuchstaben E dargestellt.

1234567890000 wird angezeigt als 1.23456789 E 12
.000000000001 wird angezeigt als 1.E − 12

Wer mit dieser Art von Notation nicht vertraut ist, sollte einmal einige sehr kleine oder sehr große Zahlenwerte eingeben, um zu beobachten, wie diese dargestellt werden.

Grenzen

Die größte Zahl, mit der der PC-1360 operieren kann, besteht aus 10 Stellen plus 2 Stellen im Exponenten. Mit anderen Worten die größte Zahl ist

9.999999999 E 99 = 99999999990000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000

und die kleinste Zahl ist

9.999999999 E −99 = .00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000009
999999999

Werte-Schablone

In einer einfachen Kettenrechnung kann das Ergebnis der vorherigen Berechnung lediglich als erstes Zahl der gegenwärtigen Berechnung verwendet werden. Zum Beispiel:

Eingabe	Anzeige
3 + 4 ENTER	3+4 7.
* 5	7.*5_
ENTER	35.

Der PC-1360 hat aber eine Funktion, die Ihnen erlaubt, das Ergebnis der vorherigen Berechnung zurückzurufen und es irgendwo in der gegenwärtigen Berechnung zu benutzen. Dies nennt man eine Werte-Schablone und der vorherige Wert kann x-beliebig mal zurückgerufen werden, indem Sie die ↓- oder ↑-Taste drücken. Wenn Sie das letzte Beispiel eingegeben haben, so drücken Sie CLS und dann entweder ↓ oder ↑ und die "35" wird angezeigt.

Schauen wir uns ein Beispiel an, wo ein vorheriges Ergebnis zweimal in der gegenwärtigen Berechnung eingesetzt wird. Bitte beachen Sie, daß sich in diesem Beispiel die Werte-Schablone verändert und auf den neuesten Stand der gegenwärtigen Berechnung gebracht wird, jedesmal, wenn Sie ENTER drücken.

Beachten Sie, daß in diesem Beispiel die letzte Antwort bei jedem Drücken der Taste ENTER durch die aktuelle Anatwort ersetst wird.

(Beispiel) Verwende das Ergebnis (6.25) der Operation, 50÷8, um folgendes zu berechnen: $12\times5\div6.25+24\times3-6.25=$:

Eingabe	Anzeige
50 / 8 ENTER	50/8 6.25 Letzte Antwort
12 * 5 / ↑ (or ↓)	12*5/6.25_ Zuletzt abgerufene Anatwort

+ 24 * 3 / ↑ (or ↓)

```
12*5/6.25+24*3/6.25_
          Zuletzt abgerufene Antwort
```

ENTER

```
21.12
```

CLS ↓

```
21.12_
```

Durch Ausführen einer manuellen Berechnung mit der ENTER Taste wird die letzte Antwort durch das Ergebnis der Vorangegangenen Rechnung ersetzt.

Die letzte Anatword wird durch Betätigen der CLS Taste nicht gelöscht.

Die letzte Antwort kann nicht angerufen werden, während sich der Computer im RUN-Modus befindet, die Programmausführung vorübergehend unterbrochen ist oder der Trace-Modus gewählt ist.

Maximale Rechenlänge

Die Länge der Rechnung, die Sie eingeben können, ist begrenzt auf 79 Tastenbedienungen, bevor die ENTER -Taste gedrückt wird. Wenn Sie diese Grenze überschreiten, beginnt der Cursor an zu blinken, um anzugeben, daß die folgende Eingabe ungültig ist. In diesem Fall muß die Rechnung in zwei oder mehrere Schritte aufgeteilt werden.

Wissenschaftliche Rechenoperationen

Der PC-1360 ist mit den in dieser Tabelle angegebenen mathematischen Basisfunktionen ausgestattet. Beachten Sie bitte, daß die Darstellung der Funktionen in BASIC sich von der normalen mathematischen Schreibung unterscheiden kann.

Funktion	Übliche Schreibweise	Tastenoperation	Bemerkungen
Trigonometrische Funktionen	sin cos tan	SIN S TAN	
Trigonometrische Umkehrfunktionen	sin $^{-1}$ cos $^{-1}$ tan $^{-1}$	ASN ACS ATN	
Zehnerlogarithmus	log	LOG	$\log_{10}$ X (Logarithmus zur Basis 10)
Natürlicher Logarithmus	ln	LN	$\log_e$ X (Logarithmus zur Basis E)
Exponentialfunktion	e^x	EXP	e=2,718281828
Potenzierung		^	A^{B}for A^B
Quadratwurzel	$\sqrt{}$	$\sqrt{}$ or SQR	
Umwandlung Dezimal-Grade nach Winkelgrade (Grade, Mi-nuten, Sekunden)		DMS	Winkleumwandlung (die Ø darf nicht wie in DEG.5 wegfallen)
Umwandlung Winkel-Grade (Grade, Minuten, Sekunden) nach Dezimal-Grade		DEG	
Ganze Zahlen		INT	INT(x) ergibt die größte ganze Zahl kleiner oder gleich x
Betrag	|X|	ABS	S ergibt den Betrag von x
Vorzeichen		SGN	Ergibt 1, wenn x>Ø, −1, wenn x<Ø und Ø, wenn x=Ø für SGN (x)
PI	π	π or PI	π=3,141592654
Hexadezimal → dezimale Schreibweise		&	Wandelt eine hexadezimale Zahl x auf der Basis 10 (&x) um.

Winkeinheit	Befehl	Erklärung
Alt-Grad Bogenmaß Neugrad	DEGREE RADIAN GRAD	Rechter Winkel = 90(°) Rechter Winkel = π/2(rad). Rechter Winkel = 100(g).

Diese Befehle werden benutzt, um Winkelmaße in Programmen zu spezifizieren. Benutzen Sie diese Befehle zur Übung in den folgenden Rechenbeispielen:

(Beispiel) sin 3Ø =

(Operation) DEGREE **ENTER** (spezifiziert "Alt-Grad" als Win5keleinheit)

SIN 3Ø **ENTER** | Ø.5

(Beispiel) $\tan \frac{\pi}{4} =$

(Operation) RADIAN **ENTER** (spezifiziert "Bogenmaß" als Winkleinheit)

TAN (PI/4) **ENTER** | **1.**

(Beispiel) $\cos^{-1}(-0.5) =$

(Operation) DEGREE **ENTER** (spezifiziert "Alt-Grad" als Winkeleinheit)

ACS −0.5 **ENTER** | **120.**

(Beispiel) log 5 + ln 5 =

(Operation) LOG5 + LN5 **ENTER** | **2.308407917**

(Beispiel) $e^{2+3} =$

(Operation) EXP (2 + 3) **ENTER** | **148.4131591**

(Beispiel) $\sqrt{4^3 + 6^4} =$

(Operation) √ (4^3+6^4) **ENTER** | **36.87817783**

(Beispiel) Wandle 30 Grad 30 Min. in DMS-Darstellung um in Dezimaldarstellung.

(Operation) DEG 30.30 **ENTER** | **30.5**
(30.5 degree)

(Beispiel) Wandle 30.755 Grad in Dezimaldarstellung um in DMS-Darstellung

(Operation) DMS 30.755 **ENTER** | **30.4518**
(30deg. 45 min. 18 sec.)

(Beispiel) Wandle CF8 in eine Dezimalzahl um.

(Operation) &CF8 **ENTER** | **3320.**

Rangfolge beim Rechnen ohne Programmunterstützung

Sie können Ausdrücke in derselben Reihenfolge eingeben, in der sie geschrieben sind, einschließlich der Klammerung oder der Funktionen. Die interne Rangfolge bei der Berechnung und die Behandlung von Zwischenergebnissen werden vom Computer selbst kontrolliert.

Die interne Rangfolge beim Rechnen ohne Programmunterstützung ist die folgende:

1) Abruf von Variablen oder Pi
2) Funktionen (sin, cos usw.)
3) Exponentiation (^)
4) Vorzeichen (+, −)
5) Multiplikation oder Division (*, /)
6) Addition oder Subtraktion (+, −)
7) Größenvergleich (>, >=, <, <=, <>, =)
8) Logisches AND, OR, NOT

Anmerkungen:

- Werden in einem Ausdruck Klammern verwendet, hat die innerhalb der Klammern angegebene Operation höchste Priorität.
- Zusammengesetzte Funktionen werden von rechts nach links abgearbeitet (sin cos^{-1} Ø.6).
- Mehrstufige Exponentiation (3^{4^2} or 3 ^ 4 ^ 2) wird von rechts nach links abgearbeitet.
- Von den obigen Punkten 3) und 4) hat die letzte Eingabe die höhere Priorität
  (z.B.) $-2 \wedge 4 \rightarrow -(2^4)$
  $3 \wedge -2 \rightarrow 3^{-2}$

Ausdrucken von Rechenergebnissen

Die Berechnungen und Ergebnisse können durch Betätigen von SHIFT und P←→NP ausgedruckt werden, sofern der optionale Drucker (CE-126P oder CE-140P) angeschlossen und eingeschaltet ist. (PRINT-Modus)

Wenn der Ausdruck nicht gewünscht wird, schalten Sie wahlweise die Zusatzeinrichtung CE-126P oder CE-140P aus oder drücken SHIFT und P←→NP erneut. (NON PRINT-Modus)

Achtung
Rechenfehler
Folgende Fehlertypen treten bei gewöhnlichen Rechnern, Taschen- und Personalcomputern auf:

(1) Fehler, die mit der Anzahl der Kommastellen zusammenhängen
Normalerweise ist die maximale Anzahl der Dezimalstellen, mit denen ein Computer rechnen kann, festgelegt. Beispiel: 4 ÷ 3 ergibt 1,3333333333... Bei einem Computer mit einem Maximum von 8 Stellen werden diese 8 Ziffern als signifikant bezeichnet, darüberhinausgehende Ziffern werden entweder ignoriert oder gerundet.

(Beispiel) Computer mit 10 signifikanten Stellen

4/3 ENTER → 1.33333333 33...
(10 signifikante Ziffern; 33... ignoriert, gerundet)

Aus diesem Grunde weicht das errechnete Ergebnis von dem tatsächlichen Ergebnis durch den ignorierten oder gerundeten Betrag ab. (Dieser Unterschied macht dann die Abweichung aus.)

Der Rechner ermittelt ein 12stelliges Resultat. Das Ergebnis wird gerundet und so verarbeitet, daß die Fehlerquote so weit als möglich minimiert wird und das angezeigte Ergebnis dem tatsächlichen weitgehend nahekommt.

(Beispiel) 4 ÷ 3 × 3

(1) 4/3 ✱3 ENTER → 4. — wird sukzessive verarbeitet

(2) 4/3 ENTER → 1.333333333
✱3 ENTER → 3.999999999 — wird unabhängig verarbeitet

Während der Rechner die Operation sukzessive verarbeitet, erhält er das Ergebnis mit 12 Kommastellen, die er als Resultat akzeptiert und anschließend rundet. Wenn der Rechner unabhängig arbeitet, wird der angezeigte Wert (10 Kommastellen) für die Berechnung benutzt.

(2) Fehler, die bei funktionsbestimmenden Algorithmen auf4treten
Der Computer benutzt eine Reihe von Algorithmen, um die Werte von Funktionen zu berechnen, wie z.B. Potenz- und trigonometrische Funktionen. Wenn in Rechnungen solche Funktionen eingesetzt werden, summiert sich die Fehlerquelle. Dieser Fehlerfaktor nimmt zu, je öfters diese Funktionen in der Rechnung eingesetzt werden. Der eigentliche Fehler für jede einzelne Funktion hängt von den eingesetzten Werten ab und ist besonders schwerwiegend bei singulären Stellen und Wendepunkten (d.h., wenn ein Winkel sich 90 Grad annähert, nähert sich die Tangente der Unendlichkeit). Ein Beispiel eines Algorithmus ist hier unten für Potenzrechnungen angegeben.

(Beispiel) 60^6 =

60 ^ 6 **ENTER** → 4.665599999E10

Obwohl 60^6 4.6656 $\times 10^{10}$ ergibt, geht der Rechner von einem Wert (y^x) aus, der dem folgenden Ausdruck entspricht:

$y^x = 10^{x \times \log y}$

Mit anderen Worten: 60^6 wird als $10^{6 \times \log 60}$ berechnet.

4 BEGRIFFE UND AUSDRÜCKE DES BASIC

In diesem Kapitel wollen wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennenlernen.

Zeichenfolgen-Konstanten

Der PC-1360 ist in der Lage, außer Zahlen auch Buchstaben und spezielle Symbole in vielfacher Weise zu verarbeiten. Diese Buchstaben, Zahlen und speziellen Symbole werden Zeichen genannt.

Folgende Zeichen stehen dem PC-1360 zur Verfügung:

1 2 3 4 5 6 7 8 9 Ø
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
! " # $ % & ( ) * + , − . / : ; < = > ? @ √ π ^

Im BASIC wird eine Gruppe von Zeichen als Zeichenfolge bezeichnet. Damit der PC-1360 den Unterschied zwischen einer Zeichenfolge und anderen Programmteilen, wie z.B. Befehlen oder Variablen-Bezeichnungen, erkennen kann, muß man die Zeichen, die zu einer Zeichenfolge gehören, in Anführungszeichen (") einschließen.

Es folgen einige Beispiele für Zeichenfolgen-Konstanten:

"HELLO"
"Goodbye"
"SHARP COMPUTER"

Die folgenden Beispiele werden als Zeichenfolgen-Konstanten nicht akzeptiert:

"COMPUTER	Anführungszeichen am Ende fehlen.
"ISN"T"	Anführungszeichen dürfen nicht innerhalb einer Zeichenfolge benutzt werden.

Hexadezimalzahlen

Das Dezimalsystem ist nur eines von verschiedenen Zahlensystemen. Ein anderes, dessen Bedeutung im Zusammenhang mit Computern stark angewachsen ist, ist das Hexadezimalsystem. Das Hexadezimalsystem basiert auf der Zahl 16 statt auf der 10. Um hexadezimale Ziffern zu schreiben, benutzt man die Ziffern 0 ~ 9 sowie 6 weitere "Ziffern": A, B, C, D, E und F. Diese entsprechen den Zahlen 10, 11, 12, 13, 14, 15 und 16. Wenn Ihr PC-1360 eine Zahl als hexadezimal auffassen soll, setzen Sie ein UND-Zeichen (&) vor den betreffenden Ausdruck.

```
&A    = 10
&10   = 16
&100  = 256
&FFFF = 65535
```

Variablen

Computer sind aufgebaut aus einer Vielzahl von kleinsten Speichereinheiten, genannt Bytes. Jedes Byte kann man sich als einzelnes Zeichen vorstellen. Das Wort "Byte" erfordert beispielsweise vier Speicherbytes, weil es vier Buchstaben enthält. Um herauszufinden, wieviele Bytes zum Arbeiten zur Verfügung stehen, geben Sie einfach MEM **ENTER** ein. Die angezeigte Zahl gibt an, wieviele Bytes zum Programmieren frei sind. Dieses Verfahren funktioniert prima für Worte, ist aber sehr unzulänglich beim Speichern von Zahlen. Aus diesem Grunde werden Zahlen in codierter Form gespeichert. Aufgrund dieser Codierung ist der Computer in der Lage, auch lange Zahlen in nur 8 Bytes zu speichern. Die größtmögliche Zahl, die gespeichert werden kann, ist +9.999999999 E +99.

Die kleinste Zahl ist +1.E−99. So erhalten Sie einen recht großen Zahlenbereich, innerhalb dessen Sie arbeiten können. Wenn jedoch das Ergebnis einer Rechnung diesen Rahmen übersteigt, wird der Computer Ihnen dies mitteilen, indem er eine Fehlermeldung auf dem Display ausgibt. Eine Erklärung dieser Fehlermeldung finden Sie in Anhang A. Geben Sie nun zur Probe ein:

```
9 E 99*9 ENTER
```

Damit der Computer wieder normal weiterarbeiten kann, brauchen Sie lediglich die CLS-Taste zu drücken. Aber wie speichert man denn nun so viel Information? Das ist ganz einfach. Der Computer zieht es vor, verschiedenen Datengruppen Namen zu geben. Wir wollen die Zahl 556 abspeichern. Sie können diese Zahl nennen, wie Sie möchten, aber für dieses Beispiel wollen wir ihr den Namen R geben. Die Anweisung LET kann benutzt werden, um den Computer zu veranlassen, einer Variablen einen Wert zuzuweisen, allerdings nur in einem Programmbefehl. Der LET-Befehl ist aber nicht unbedingt er-forderlich, darum werden wir ihn nicht sehr oft benutzen. Geben Sie nun R = 556 ein und drücken Sie ENTER. Der Computer hat nun den Wert 556 mit dem Buchstaben R in Verbindung ge-bracht. Diese Buchstaben, die man benutzt, um Information zu speichern, nennt man Variable. Um den Inhalt der Variablen R abzufragen, drücken Sie die CLS-Taste, dann die Tasten R und ENTER. Der Computer antwortet, indem er den Wert 556 rechts im Display anzeigt. Diese Möglichkeit kann sehr nützlich sein, wenn man Programme oder Formeln eingibt.

Wir wollen als nächstes die Variable R in einer einfachen Formel verwenden. In dieser Formel steht R für den Radius eines Kreises, dessen Fläche wir berechnen wollen. Die Formel für die Kreisfläche lautet: $A = \pi * R^2$, Geben Sie ein: R SHIFT ^ 2 * SHIFT π ENTER. Das Ergebnis ist 971179.3866. Dieses Verfahren des Gebrauchs von Variablen wird eingehender erklärt, wenn wir uns mit dem Programmieren beschäftigen.

Bislang haben wir uns nur mit numerischen Variablen befaßt. Wie speichert man nun alphabetische Zeichen? Grundsätzlich ist das Prinzip das gleiche, aber damit der Computer den Unterschied zwischen den beiden Variablentypen erkennen kann, muß nun ein $ zum Namen der Variablen gesetzt werden. Wir wollen z.B. das Wort BYTE unter der Variablen B$ speichern. Sehen Sie das $-Zeichen hinter dem B?

Dies sagt dem Computer, daß der Inhalt der Variablen B alphanumerisch bzw. eine Datenzeichenfolge ist.

Damit dies klarer wird, geben Sie ein: B SHIFT $ = SHIFT " BYTE SHIFT " ENTER. Der Wert BYTE ist nun unter der Variablen B$ gespeichert. Um sich zu vergewissern, drücken Sie die CLS-Taste und geben dann ein: B SHIFT $ ENTER. In der Anzeige steht BYTE, diesmal aber auf der linken Seite der Anzeige.

Beachten = Sie:
Der Inhalt von Zeichenfolgen oder Variablen wird von der linken Ecke der ersten Zeile an angezeigt.

Die Variablen, mit denen der SHARP PC-1360 arbeitet, sind folgendermaßen aufgegliedert:

- Variablen
  - numerische Variablen
    - vorgegebene numerische Variablen (A bis Z)
    - einfache numerische Variablen (AB,C1,usw.)
    - numerische Feld-Variablen
  - Zeichenfolgen-Variablen
    - vorgegebene Zeichenfolgen-Variablen (A$ bis Z$)
    - einfache Zeichenfolgeb-Variablen (BB$, C2$,usw.),
    - Zeichenfeld-Variablen

Vorgegebene Variablen

Der erste Bereich, die vorgegebene Variable, wird vom Computer grundsätzlich benutzt, um Daten zu speichern. Man kann ihn als reservierten Bereich denken. Mit anderen Worten, egal, wieviel Speicherplatz Ihr Programm in Anspruch nimmt, Ihnen stehen immer mindestens 26 Variablen zur Speicherung von Daten offen. Diese Daten können entweder NUMERISCH oder in ZEICHENFOLGEN(Buchstaben)-Form gewählt werden. Vorgegebene Speicherplätze haben eine Kapazität von 8 Byte und können immer nur einen Datentyp zur Zeit aufnehmen. Geben Sie zur Erklärung ein:

```
A = 123 ENTER
A$ ENTER
```

Sie erhalten die Meldung:

```
ERROR 9
```

Das bedeutet, daß Sie numerische Daten in den Speicherbereich mit dem Namen A gelegt haben und daß Sie dann dem Computer befohlen haben, diese Information als Zeichenfolge-Daten wieder abzurufen. Das aber verwirrt den Computer, so daß er eine Fehlermeldung ausgibt. Drücken Sie die **CLS**-Taste, um die Fehlermeldung zu beseitigen. Versuchen Sie nun das folgende Beispiel:

```
A$="ABC" ENTER
A ENTER
```

Wieder ist der Computer verwirrt und meldet ERROR 9.

Einfache Variablen

Einfache Variablenbezeichnungen werden durch zwei (oder mehr) alphanumerische Zeichen charakterisiert, wie z.B. AA oder B1. Anders als feste Variablen haben die einfachen Variablen keinen fest im Speicher reservierten Bereich. Der Speicherbereich für einfache Variablen wird automatisch (im Programm oder im Datenbereich) bereitgestellt, sobald eine einfache Variable erstmalig benutzt wird.

Da für einfach numerische und einfache Zeichenfolgen-Variablen verschiedene Speicherbereiche vorgesehen sind, können Variablen mit dem gleichen Namen (wie z.B. AB und AB$) gleichzeitig benutzt werden.

Während alphanumerische Bezeichnungen für einfache Variablen-Namen verwendet werden können (wie bei alphabetischen Zeichen können nur Großbuchstaben benutzt werden), muß man als erstes Zeichen eines Variablennamens grundsätzlich einen Buchstaben wählen. Werden mehr als zwei Buchstaben zur Definition eines Variablennamens benutzt, sind nur die beiden ersten Zeichen von Bedeutung.

Achtung:

- Die Namen von Funktionen oder BASIC-Befehlen dürfen für den PC-1360 nicht zur Bezeichnung von Variablen benutzt werden (z.B.) PI, IF, TO, ON, SIN usw.
- Unter einer einzelnen Zeichenfolgen-Variablen können bis zu 16 Zeichen oder Symbole gespeichert werden.

Feld-Variablen

In einigen Fällen ist es sinnvoll, Zahlen in organisierten Gruppen zu verarbeiten, z.B. eine Tabelle der Fußball-Ergebnisse oder eine Steuertabelle. im BASIC werden solche Gruppen Felder genannt. Ein Feld kann eindimensional sein, wie z.B. eine Liste, es kann aber auch zweidimensional sein, wie z.B. eine Tabelle.

Um ein Feld zu definieren, benutzt man den DIM-Befehl (Kürzel für Dimension). Felder müssen vor Gebrauch immer definiert werden. (Dies war nicht der Fall bei den Einwert-Variablen, die wir bislang benutzt haben.) Die Form für die DIMensionierung numerischer Felder ist:

DIM variablenname (größe)

Dabei bedeutet:

variablenname ist eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für numerische Variablen.

Größe ist eine Anzahl der Speicherplätze, die sich im Bereich zwischen Ø und 255 bewegen muß. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, ein Speicherplatz mehr bereitgestellt wird.

Beispiele für mögliche DIMensionierungsbefehle sind:

```
DIM X (5)
DIM AA (24)
DIM Q5 (Ø)
```

Der erste Befehl schafft ein Feld X mit 6 Speicherplätzen. Der zweite Befehl baut ein Feld mit 25 Speicherplätzen auf, der dritte ein Feld Q5 mit einem Speicherplatz, was im Grunde unsinnig ist (zumindest für Zahlen), da man ebensogut eine einwertige numerische Variable definieren könnte.

Es ist wichtig zu wissen, daß eine Feld-Variable X und eine Variable X sich unterscheiden und auch vom SHARP PC-1360 unterschieden werden können. Das erste X bezeichnet eine Serie von numerischen Speicherplätzen, das zweite einen einzelnen und getrennten Speicherplatz.

Nachdem Sie nun wissen, wie man Felder aufbaut, mögen Sie sich fragen, wie man die einzelnen Speicherplätze anspricht. Da die gesamte Gruppe unter einem einzigen Namen abgelegt ist, sprechen wir einen einzelnen Speicherplatz ("Element" genannt) an, indem wir an den Namen der Gruppe eine Zahl in Klammern anschließen. Diese Zahl wird "Index" genannt. So müßte man z.B., um die Zahl 8 an fünfter Stelle in unserem (vorher definierten) Feld X unterzubringen, schreiben:

```
X(4) = 8
```

Wenn Sie die Zahl 4 verwirrt, bedenken Sie, daß die Nummerierung der Elemente in einem Feld mit Null beginnt und dann bis zu der in der DIM-Anweisung definierten Anzahl von Elementen fortläuft.

Der besondere Vorteil von Feldern liegt in der Möglichkeit, einen längeren Ausdruck oder eine Variable als Index zu benutzen.

Um ein Zeichenfeld zu DIMensionieren, muß man etwas anders vorgehen als bei numerischen Feldern:

DIM zeichenvariablenname (Größe) * Länge

Dabei bedeutet:

zeichenvariablenname ist eine Bezeichnung der Variablen gemäß den oben besprochenen Benennungsregeln für normale Zeichenvariablen.

Größe ist die Anzahl der Speicherplätze (im Bereich zwischen 0 und 255). Beachten Sie, wenn Sie eine Größenordnung angeben, daß ein Speicherplatz mehr bereitgestellt wird.

*Länge ist fakultativ. Wenn sie angegeben wird, spezifiziert sie die Länge der einzelnen, im Feld enthaltenen Zeichenfolgen. Die Länge muß im Bereich zwischen 1 und 80 liegen. Wird die Länge nicht angegeben, erhalten die Zeichenfolgen automatisch eine Länge von 16 Zeichen.

Beispiele für mögliche DIMensionierungsbefehle für Zeichenfelder sind:

```
DIM X$(4)
DIM NM$(10)*10
DIM IN$(1)*80
DIM R$(0)*26
```

Im ersten Fall wird ein Feld mit fünf Zeichenfolgen aufgebaut, in denen je 16 Zeichen gespeichert werden können. Der zweite DIM-Befehl baut ein Feld NM auf mit elf Zeichenfolgen zu je 10 Zeichen.

Definiert man auch Zeichenfolgenlängen, die kleiner als 16 Zeichen sind, spart man natürlich Speicherkapazität. Das dritte Beispiel zeigt ein Feld mit zwei Elementen zu je 80 Buchstaben, und im letzten Fall wird eine einzige Zeichenfolge von 26 Zeichen aufgebaut.

Neben den einfachen Feldern, die wir gerade behandelt haben, läßt der PC-1360 auch "zweidimensionale" Felder zu. Ein eindimensionales Feld ist eine Liste von Daten, die in einer einzigen Spalte angeordnet sind. Entsprechend ist ein zweidimensionales Feld eine Tabelle mit Zeilen und Spalten.

Zweidimensionale Felder werden folgendermaßen DIMensioniert:

DIM numerischer Variablenname (zeilen, spalten)

oder

DIM zeichenvariablenname (zeilen, spalten)*länge

Dabei bedeutet:

Zeilen gibt die Anzahl der Zeilen im Feld an. Dies muß eine Zahl zwischen Ø und 255 sein. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, eine Zeile mehr bereitgestellt wird.

Spalten gibt die Anzahl der Spalten im Feld an. Dies muß eine Zahl zwischen 0 und 255 sein. Beachten Sie, daß, wenn Sie eine Größenordnung eingeben, eine Spalte mehr eingerichtet wird.

Die folgende Tabelle illustriert die Speicherplätze, die sich aus der Anweisung DIM T (2,3) und den (jetzt aus zwei Zeichen bestehenden) Indexen, die zu den jeweiligen Speicherplätzen gehören, ergibt:

	Spalte Ø	Spalte 1	Spalte 2	Spalte 3
Zeile Ø	T (Ø, Ø)	T (Ø, 1)	T (Ø, 2)	T (Ø, 3)
Zeile 1	T (1, Ø)	T (1, 1)	T (1, 2)	T (1, 3)
Zeile 2	T (2, Ø)	T (2, 1)	T (2, 2)	T (2, 3)

Achtung:
Zweidimensionale Felder nehmen viel Speicherplatz in Anspruch. Z.B. benötigt ein Feld mit 25 Zeilen und 35 Spalten 875 Speicherplätze!

Felder sind sehr nützliche Programmier-Hilfsmittel.

Die folgende Tabelle zeigt die Anzahl der Bytes, die zur Definierung jeder einzelnen Variable benötigt werden sowie die Anzahl der für jeden einzelnen Programm-Befehl erforderlichen.

Variable	Variablenname	Daten	
Numerische Variable	7 Bytes	8 Bytes	
Zeichenfolge-Variable	7 Bytes	Feldvariable	Spezifizierte Anzahl
		Einfache Variable (2-Zeichen-Variable)	16 Bytes

* Wenn z.B. DIM Z$ (2, 3)*10 spezifiziert wird, werden 12 Variablen mit einer Speicherkapazität von je 10 zeichen bereitgestellt. Dies erfodert 7 Bytes (Variablenname) + 10 Bytes (Anzahl der Zeichen) × 12 = 127 Bytes.

Element	Zeilennummer	Befehl & Funktion	ENTER, und andere
Anzahl der verwendeten Bytes	3 Bytes	2 Bytes	1 Bytes

Variablen der Form A( )

Wird im Speicher des Computers ein Bereich für vorgegebene Variablen reserviert, kann dieser auch benutzt werden, um mit einem Index versehene Variablen, die die gleiche Form wie Feld-Variablen haben, zu definieren.

26 vorbestimmte Variablennamen stehen zur Verfügung, nämlich A bis Z (A$ bis Z$). Jeder dieser Namen kann als Index eine der Zahlen von 1 bis 26 tragen, wie z.B. A(1)—A(26) oder A$(1)—A$(26). Das bedeutet, daß die Variable A(1) anstelle der Variablen A benutzt werden kann, A(2) anstelle von B, A(3) anstelle von C usw.

In der folgenden Abbildung können Sie sehen, daß die Variable A im selben Speicherbereich liegt wie die Variable A$, daß B sich mit B$ deckt usw. für alle Buchstaben des Alphabets.

Abbildung:

A = A$ = A(1) = A$(1)
B = B$ = A(2) = A$(2)
C = C$ = A(3) = A$(3)
D = D$ = A(4) = A$(4)
E = E$ = A(5) = A$(5)
F = F$ = A(6) = A$(6)
G = G$ = A(7) = A$(7)
H = H$ = A(8) = A$(8)
I = I$ = A(9) = A$(9)
J = J$ = A(10) = A$(10)
K = K$ = A(11) = A$(11)
L = L$ = A(12) = A$(12)
M = M$ = A(13) = A$(13)
N = N$ = A(14) = A$(14)
O = O$ = A(15) = A$(15)
P = P$ = A(16) = A$(16)

```
Q = Q$ =  A(17) =  A$(17)
R = R$ =  A(18) =  A$(18)
S = S$ =  A(19) =  A$(19)
T = T$ =  A(Ø) =  A$(Ø)
U = U$ =  A(21) =  A$(21)
V = V$ =  A(22) =  A$(22)
W = W$ =  A(23) =  A$(23)
X = X$ =  A(24) =  A$(24)
Y = Y$ =  A(25) =  A$(25)
Z = Z$ =  A(26) =  A$(26)
```

Wenn allerdings schon ein Feld mit dem Namen A oder A$ durch einen DIM-Befehl definiert wurde, können Index-Variablen mit den Namen A nicht zusätzlich definiert werden. Wurde beispielsweise ein Feld A definiert durch DIM A(5), sind die Speicherplätze A(Ø) bis A(5) im Programm-/Datenbereich reserviert. Wenn man dann eine Variable A(2) spezifiziert, bezieht sich diese nicht auf die vorbestimmte Variable B, sondern auf die Feld-Variable A(2) im Programm-/Datenbe4reich. Spezifizieren Sie A(9), wird dies eine Fehlermeldung bewirken, da A(9) außerhalb des durch den DIM A(5)-Befehl bestimmten Bereiches liegt.

Wenn andererseits Index-Variablen der Form A( ) eingegeben worden sind, ist es nicht möglich, gleichzeitig Felder A oder A$ mit Hilfe eines DIM-Befehls einzurichten, es sei denn, die Definition der Index-Variablen wird mit dem CLEAR-Befehl gelöscht.

Sollen Indexen größer als 26 für Variablen der Form A( ) benutzt werden, wenn ein Feld A nicht durch ein DIM-Befehl definiert wurde, werden die entsprechenden Speicherplätze im Programm-/Datenbereich für diese A ( )-Variablen definiert. Wenn Sie z.B. A(35) = 5 eingeben, werden die Speicherplätze A(27) bis A(35) im Programm-/Datenbereich reserviert.

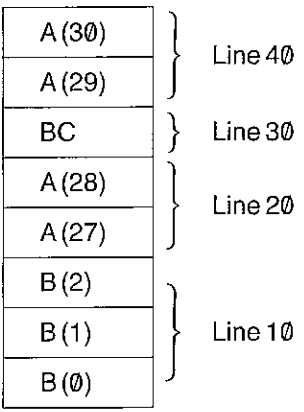
Da Variablen mit einem Index größer als 26 wie Feld-Variablen behandelt werden, unterliegen sie den folgenden besonderen Beschränkungen:

(1) Die Speicherplätze für ein Feld mit demselben Namen müssen im Programm-/Datenbereich nebeneinander liegen. Andernfalls erhalten Sie eine Fehlermeldung.

```
10: DIM B(2)
20: A(28)=5
30: BC=12
40: A(30)=9
```

Wenn dieses Programm eingegeben wird, befinden sich die Definitionen für das Feld "A" nicht in zwei aufeinanderfolgenden Segmenten des Programm-/Datenbereiches und aus Zeile 40 wird sich ein Fehler ergeben.

[Programm-Datenbereich]



(2) Numerische Feld-Variablen und Zeichenfolgen-Feld-Variablen mit demselben Index können nicht zur gleichen Zeit definiert werden. So können z.B. A(30) und A$(30) nicht zur gleichen Zeit definiert werden, da sie im Programm-/ Datenbereich den gleichen Platz einnehmen.

(3) Zweidimensionale Felder können nicht definiert werden, ebensowenig ist es möglich, die Länge der Zeichenfolgen zu definieren, die in einer Feld-Variable enthalten sein sollen. Die Länge der Buchstabenzeichenfolge, die von der Variablen A$( ) erfaßt werden kann, ist beispielsweise auf 7 oder weniger Zeichen beschränkt.

(4) Variablen mit dem Index (0) können nicht definiert werden. Wenn Sie A(0) oder A$(0) definieren, werden Sie eine Fehlermeldung erhalten.

(5) Wenn A(27) oder A$(27) und größer als erste benutzt werden, werden 7 Bytes für den Variablennamen und 8 Bytes für jede Variable belegt.

Ausdrücke

Ein Ausdruck ist eine Kombination von Variablen, Konstanten und Operatoren, die durch Rechenoperationen auf einen einzigen Wert zusammengefaßt werden kann. Die Rechenbeispiele, die Sie in Kapitel 3 eingegeben haben, waren Beispiele für solche Ausdrücke. Ausdrücke sind ein wesentlicher Bestandteil von BASIC-Programmen. Z.B. kann ein Ausdruck eine Formel sein, mit der das Ergebnis einer Gleichung ermittelt wird, oder ein Test zur Bestimmung des Verhältnisses zwischen zwei Größen, ein Mittel, um eine Reihe von Zeichenfolgen zu formatieren.

Numerische Operatoren

Der PC-1360 verfügt über fünf numerische Operatoren. Dies sind die arithmetischen Rechenzeichen, die Sie verwendet haben, als Sie die Benutzung des PC-1360 als Taschenrechner in Kapitel 3 kennenlernten.

\+ Addition
\- Subtraktion
* Multiplikation
/ Division
^ Exponentiation

Ein numerischer Ausdruck wird in derselben Weise konstruiert, wie Sie komplexe Rechenbefehle eingegeben haben. Numerische Ausdrücke können jede aussagefähige Kombination von numerischen Konstanten, numerischen Variablen und numerischen Operatoren beinhalten:

```
(A*B)^2
A(2,3)+A(3,4)+5.0-C
(A/B)*(C+D)
```

Zeichenfolgen-Ausdrücke

Zeichenfolgen-Ausdrücke sind ähnlich den numerischen Ausdrücken, allerdings gibt es nur einen einzigen Zeichenfolgen-Operator: die Verkettung (+). Das benutzte Symbol ist dasselbe wie das Plus-Zeichen. Wird es mit einem Zeichenfolgen-Paar benutzt, knüpft das + die zweite Zeichenfolge an das Ende der ersten an und schafft dadurch eine längere Zeichenfolge. Wenn Sie komplexere Zeichenfolgen-Verkettungen und andere Zeichenfolgen-Operationen vornehmen, bedenken Sie bitte, daß der PC-1360 nur 80 Zeichen zur Zeit annimmt.

Achtung:
Zeichenfolgen-Einheiten und numerische Einheiten können nicht in demselben Ausdruck definiert werden, es sei denn, man benutzt eine der Funktionen, die Zeichenfolgen-Werte in numerische Werte umwan4deln oder umgekehrt.

"15" + 1Ø	ist unzulässig
"15" + "1Ø"	ist "151Ø", nicht "25"

Verhältnis-Ausdrücke

Ein Verhältnis-Ausdruck vergleicht zwei Ausdrücke und gibt an, ob das festgestellte Verhältnis wahr oder unwahr ist. Die Verhältnis-Operatoren sind:

>	größer als
>=	größer oder gleich
=	gleich
<>	ungleich
<=	kleiner oder gleich
<	kleiner als

Die folgenden Ausdrücke werden als Verhältnis-Ausdrücke akzeptiert:

A < B
C(1,2) >= 5
D(3) <> 8

Wenn A gleich 1Ø wäre, B gleich 12, C(1,2) gleich 6 und D(3) gleich 9, wären alle diese Ausdrücke wahr.

Zeichenfolgen können ebenfalls mit Hilfe von Verhältnis-Ausdrücken verglichen werden. Die beiden Zeichenfolgen werden für Zeichen gemäß dem Wert ihres ASCII-Codes verglichen (siehe Anhang B: ASCII-Code-Tabelle). Ist eine Zeichenfolge kürzer als die andere, wird 0 oder NUL in die freibleibenden Positionen eingesetzt. Alle folgenden Beispiele sind wahr:

```
"ABCDEF" = "ABCDEF"
"ABCDEF" <> "ABCDE"
"ABCDEF" > "ABCDE"
```

Verhältnis-Ausdrücke beurteilen nach wahr oder unwahr. Beim PC-1360 wird wahr durch 1 angegeben, unwahr durch ein 0. In jeder logischen Überprüfung wird ein Ausdruck, der 1 oder mehr ergibt, als wahr betrachtet, während jeder, der Ø ergibt, als unwahr verstanden wird. Gute Programmierpraktiken verlangen aber den Gebrauch eines expliziten Verhältnis-Ausdrucks statt sich auf diese Koinzidenz zu verlassen.

Logische Ausdrücke

Logische Ausdrücke sind Verhältnis-Ausdrücke, die die Operatoren AND, OR und NOT benutzen. AND und OR werden verwendet, um zwei Verhältnis-Ausdrücke miteinander zu verbinden; der Wert des verbundenen Ausdrucks wird aus der folgenden Tabelle ersichtlich:

A AND B

Wert von B \ Wert von A	wahr	Unwahr
wahr	wahr	unwahr
unwahr	unwahr	unwahr

A OR B

Wert von B \ Wert von A	wahr	unwahr
wahr	wahr	wahr
unwahr	wahr	unwahr

(Anmerkung: Der Wert von A und B muß Ø oder 1 sein)

- Dezimalstellen können folgendermaßen in binärer Schreibweise von 16 Bit dargestellt werden:

DEZIMALFORM	BINÄRFORM VON 16 BIT
32767	0111111111111111
3	0000000000000011
2	0000000000000010
1	0000000000000001
0	0000000000000000
−1	1111111111111111
−2	1111111111111110
−3	1111111111111101
−32768	1000000000000000

Die negative (NOT) Form einer binären Zahl 0000000000000001 wird folgendermaßen dargestellt:

```
NOT          0000000000000001
(Negative)──►1111111111111110
```

So wird also für jedes Bit 1 zu 0 interpretiert, und 0 zu 1, was man "Inversion (NOT)" nennt.
Demnach ergibt sich das Folgende, wenn man 1 und NOT 1 addiert:

```
    0000000000000001 (1)
+)  1111111111111110 (NOT 1)
--------------------
    1111111111111111 (−1)
```

Alle Bits werden also 1. Entsprechend der obigen Zahlenliste werden die Bits in dezimaler Schreibweise −1, d.h. 1 + NOT 1 = −1.
Das Verhältnis zwischen einem numerischen Wert X und seiner negativen Form (NOT X) ist:

X + NOT X= −1

Daraus ergibt sich die Gleichung NOT X = −X −1
d.h. NOT X = − (X + 1)
Aus dieser Gleichung lassen sich folgende Ergebnisse ableiten.

NOT Ø = −1
NOT −1 = Ø
NOT −2 = 1

Mit diesen logischen Operationen können Sie auch mehr als zwei Ausdrücke in Beziehung setzen. Dann sollten Sie aber darauf achten, Klammern zu verwenden, um den gewünschten Vergleich zu verdeutlichen.

(A<9) AND (B>5)
(C = 5) OR (C = 6) OR (C = 7)

Der PC-1360 behandelt logische Operationen "bitweise" als logische Funktionen auf der Basis von 16 Bit (siehe Anmerkung zu den Verhältnis-Ausdrücken und wahr/unwahr). In normalen Operationen hat dies keine Bedeutung, weil die einfache 1 und 0 (wahr und unwahr), die aus Verhältnis-Ausdrücken resultieren, nur je ein Bit einnehmen. Wenn Sie jedoch logische Operatoren auf andere Werte als 0 oder 1 anwenden, wird jedes Bit einzeln bearbeitet. Wenn z.B. A = 17 ist, und B = 22, dann ist (A OR B) = 23.

17 ist in binärer Schreibweise	10001
22 ist in binärer Schreibweise	10110
17 OR 22 ist	10111
	10111 in dezimaler Schreibweise ist aber 23.

Wenn Sie ein geübter Programmierer sind, kennen Sie sicherlich Anwendungen, wo diese Art der Operation sehr nützlich sein kann. Programmieranfänger sollten sich vorerst an klare, einfache Wahr-/Unwahr-Aussagen halten.

Kmerung und Vorrang der Operatoren

Bei der Bearbeitung komplexer Ausdrücke folgt der PC-1360 einer Reihe vordefinierter Prioritäten, die bestimmen, in welcher Reihenfolge die Operatoren bearbeitet werden. Das kann sehr wichtig sein:

5 + 2 * 3 kann sein

5 + 2 = 7 oder 2 * 3 = 6
7 * 3 = 21 6 + 5 = 11

Die genauen Regeln des "Operatoren-Vorrangs" werden in Anhang D beschrieben.

Damit Sie sich nicht alle diese Regeln merken müssen, und damit Sie Ihre Programme eindeutiger gestalten, benutzen Sie immer Klammern, um die Reihenfolge der Bearbeitung von Operatoren vorzugeben. Das obige Beispiel wird eindeutig, wenn Sie schreiben:

(5 + 2) * 3 oder 5 + (2 * 3)

RUN-Modus

Im allgemeinen kann jeder der oben angesprochenen Ausdrücke ebenso im RUN-Modus benutzt werden wie bei der Programmierung einer BASIC-Anweisung. Im RUN-Modus wird ein Ausdruck sofort ausgewertet und angezeigt, z.B.

Eingabe	Anzeige
(5>3) AND (2<6) **ENTER**	1.

Die 1 gibt an, daß der Ausdruck wahr ist.

Funktionen

Funktionen sind spezielle Bestandteile der BASIC-Sprache, die einen Wert in einen anderen Wert transformieren. Funktionen arbeiten wie Variablen, deren Wert durch andere Variablen oder Ausdrücke festgelegt wurde. ABS ist eine Funktion, die den absoluten Wert der Zahl hervorbringt, auf die sie angewendet wird.

ABS(−5)	ist	5
ABS(6)	ist	6

LOG ist eine Funktion, die den dekadischen Logarithmus der Zahl berechnet, auf die LOG angewendet wird.

LOG (1ØØ)	ist	2
LOG (1ØØØ)	ist	3

Eine Funktion kann überall dort verwendet werden, wo auch Variablen benutzt werden können. Viele Funktionen erfordern keine Klammerung.

LOG 1ØØ	ist dasselbe wie	LOG (1ØØ)

Sie müssen jedoch Klammern benutzen, sobald sich eine Funktion auf mehr als eine Zahl bezieht. Der Gebrauch von Klammern sorgt immer für eine klarere und eindeutigere Programmierung.

In Kapitel 9 finden Sie die auf dem PC-1360 verfügbaren Funktionen.

5 PROGRAMMIEREN DES PC-1360

Im vergangenen Kapitel haben wir einige Begriffe und Ausdrücke der Programmiersprache BASIC kennengelernt. In diesem Kapitel wollen wir nun diese Elemente benutzen, um auf dem PC-1360 Programme zu schreiben. Wir möchten noch einmal darauf hinweisen, daß dieses Handbuch nicht als Einführung in die BASIC-Programmierung verstanden werden soll. Dieses Kapitel soll Sie lediglich in den besonderen Gebrauch des BASIC auf dem PC-1360 einführen.

Programme

Ein Programm besteht aus einer Reihe von an den Computer gerichteten Befehlen. Denken Sie daran, daß der PC-1360 nur eine Maschine ist, die genau die Operationen durchführt, die Sie angeben. Sie als Programmierer sind dafür verantwortlich, korrekte Befehle einzugeben.

BASIC-Anweisungen

Der PC-1360 setzt Programme entsprechend einem bestimmten Format um. Dieses Format wird Anweisung genannt. Sie geben die BASIC-Anweisungen immer nach einem bestimmten Muster ein. Anweisungen müssen mit einer Zeilennummer beginnen.

```
1Ø: INPUT A
2Ø: PRINT A*A
3Ø: END
```

Zeilennummern

Jede Programmzeile muß eine eigene Nummer haben, und zwar muß diese eine ganze Zahl zwischen 1 und 65279 sein. Zeilennummern sind die Bezugspunkte des Computers. Sie geben dem PC-1360 an, in welcher Reihenfolge er ein Programm abarbeiten muß. Es ist nicht erforderlich, daß Sie die Programmzeilen folgerichtig eingeben (obwohl dieser Weg sicher der weniger verwirrende ist, besonders, wenn Sie noch Anfänger sind). Der Computer beginnt beim Durcharbeiten eines Programms immer mit der niedrigsten Zeilennummer und arbeitet die folgende in aufsteigender Reihenfolge ab.

Beim Programmieren ist es sinnvoll, genug Raum für spätere Einschübe zwischen den einzelnen Zeilen zu lassen (1Ø, 2Ø, 3Ø, usw.). Dies ermöglicht den Einschub von zusätzlichen Zeilen, falls notwendig.

ACHTUNG:
Benutzen Sie keine gleichen Zeilennummern in verschiedenen Programmen, die Sie später mischen wollen.

Wenn Sie gleiche Zeilennummern benutzen, wird die ältere Zeile gelöscht, sobald Sie eine neue mit derselben Nummer eingeben.

BASIC-Befehle

Alle BASIC-Anweisungen müssen Befehle enthalten. Diese Befehle sagen dem Computer, welche Operationen er durchführen soll. Ein Befehl ist immer Programmbestandteil, insofern wird die Operation nicht direkt darauf erfolgen.

Einige Anweisungen erfordern oder erlauben den Gebrauch eines Operanden:

```
1Ø: DATA "HELLO"
2Ø: READ B$
3Ø: PRINT B$
4Ø: END
```

Operanden informieren den Computer darüber, auf welche Daten sich die vom Befehl angeordnete Operation bezieht. Einige Befehle erfordern Operanden, bei anderen Befehlen sind sie fakultativ. Einige Befehle erlauben keine Operanden. (In Kapitel 9 finden Sie die BASIC-Befehle und ihren Gebrauch auf dem PC-1360.)

Beachten Sie:
Befehle, Kommandos und Funktionen müssen in Großbuchstaben eingegeben werden.

BASIC-Kommandos

Kommandos sind Anweisungen an den Computer, die außerhalb eines Programms eingegeben werden. Kommandos fordern den Computer auf, bestimmte Handlungen mit dem Programm vorzunehmen oder einen bestimmten Modus zu setzen, der dann wiederum die Art der Programmbearbeitung determiniert. Anders als die Befehle haben Kommandos unmittelbare Wirkung - sobald Sie die Eingabe des Kommandos beendet haben (mit Betätigen der **ENTER** -Taste), wird das Kommando ausgeführt. Kommandos geht keine Zeilennummer voraus.

RUN
NEW
RADIAN

Einige Kommandos können auch als Befehle verwendet werden. (In Kapitel 9 finden Sie die BASIC-Kommandos und und ihren Gebrauch auf dem PC-1360.)

Modi

Sie erinnern sich sicherlich, daß Sie, als Sie den PC-1360 als Rechner benutzt haben, den RUN-Modus gewählt haben.

Der RUN-Modus wird ebenfalls gebraucht, um die von Ihnen geschriebenen Programme abzuarbeiten.

Der PROgramm-Modus wird gewählt, wenn Sie Programme eingeben oder aufbereiten wollen. Der RESERVE-Modus ermöglicht es, vordefinierte ZeichenfolgenVariablen zu kennzeichnen und zu speichern. Weiterhin wird er für komplexere Programmierung gebraucht (siehe Kapitel 6).

Der Anfang des Programmierens mit dem PC-1360

Nachdem Sie in vielen Übungen den PC-1360 als Taschenrechner benutzt haben, sind Sie sicherlich schon recht vertraut mit der Tastatur. Von nun an werden wir, wenn wir eine Eingabe zeigen, nicht mehr jede Tastenbedienung zeigen. Denken Sie immer daran, **SHIFT** zu betätigen, um Zugang zu den Zeichen oberhalb der Tasten zu bekommen und vergessen Sie nicht, AM ENDE JEDER ZEILE DIE **ENTER**-TASTE ZU DRÜCKEN.

Nun können Sie mit dem Programmieren beginnen! Um Programmanweisungen in den Computer einzugeben, schalten Sie den Computer erst in den PROGRAM-Modus, indem Sie die Taste **MODE** benutzen. Auf dem Display wird folgende Anzeige erscheinen:

```
PRO   PROGRAM MODE
      >
```

Geben Sie das NEW-Kommando ein.

Eingabe

NEW **ENTER**

Anzeige

```
NEW
>
```

Das NEW-Kommando löscht aus dem Arbeitsspeicher des PC-1360 alle vorhandenen Programme und Daten. Das Bereitschaftssymbol erscheint, wenn Sie die **ENTER**-Taste gedrückt haben. Es zeigt an, daß der Computer nun Eingaben erwartet.

Beispiel 1 — Eingabe und Ablauf eines Programms

Vergewissern Sie sich, daß der PC-1360 sich im PRO-Modus befindet, und geben Sie das folgende Programm ein:

Eingabe	Anzeige
1Ø PRINT "HELLO"	**1Ø:PRINT "HELLO"**

Beachten Sie, daß der PC-1360 Ihre Eingabe anzeigt, sobald Sie **ENTER** gedrückt haben, und daß er dabei einen Doppelpunkt (:) zwischen die Zeilennummer und den Befehl einfügt. Vergewissern Sie sich, daß die Anweisung in der richtigen Form eingegeben wurde.

Nun schalten Sie den Computer auf den RUN-Modus um:

Eingabe	Anzeige
CLS RUN	**HELLO**

Da dies die einzige Programmzeile ist, wird der Computer an dieser Stelle stoppen. Drücken Sie **ENTER**, um aus dem Programm herauszukommen und geben Sie RUN vor, wenn das Programm noch einmal ablaufen soll.

Beispiel 2 — Aufbereitung eines Programms

Nehmen wir an, Sie wollen den Inhalt dessen, was Ihr Programm anzeigt, verändern, d.h. Sie wollen Ihr Programm aufbereiten. Bei einem einzeiligen Programm könnten Sie die Eingabe schlicht wiederholen, aber wenn Sie komplexere Programme schreiben, wird die Aufbereitung zu einem wichtigen Bestandteil des Programmierens. Wir wollen das Programm, das Sie gerade geschrieben haben, aufbereiten.

Sind Sie noch im RUN-Modus? Wenn ja, betätigen Sie die MODE-Taste, um wieder in den PROgramm-Modus zu gelangen.

Um Ihr Programm aufbereiten zu können, müssen Sie es nun wieder abrufen. Benutzen Sie dazu die ↑-Taste. Sofern Ihr Programm vollständig durchlaufen worden war, wird die ↑-Taste lediglich die letzte Programmzeile abrufen. Falls ein Fehler im Programm vorlag oder Sie die Ausführung mit der BREAK (BRK)-Taste unterbrochen hatten, wird die ↑-Taste die Zeile abrufen, in der der Fehler liegt oder in der BREAK betätigt wurde. Um Veränderungen in Ihrem Programm vorzunehmen, bedienen Sie sich der ↑-Taste, um sich im Programm nach oben zu bewegen (d.h. die jeweils vorherige Zeile abzurufen), und der ↓-Taste, um sich nach unten zu bewegen (d.h. die jeweils nächste Zeile abzurufen). Wenn die ↑- und ↓-Tasten festgehalten werden, rollt das Programm im Display nach oben bzw. unten, d.h. die einzelnen Zeilen werden nacheinander durch das Display laufen.

Sie erinnern sich sicherlich, daß Sie den Cursor innerhalb einer Programmzeile, die jeweils in der obersten Zeile des Displays angezeigt wird, mit dem → (rechter Pfeil) und dem ← (linker Pfeil) bewegen können. Mit Hilfe des →-Pfeiles setzen Sie nun den Cursor auf das erste Zeichen, das Sie ändern möchten.

Achtung:
Obwohl auf dem Display mehrere Zeilen angezeigt werden können, kann der Cursor nur in der obersten Zeile bewegt werden. Um eine andere Zeile aufzubereiten, muß die gewünschte Zeile mit der ↓-Taste nach oben verschoben und dann aufbereitet werden.

Eingabe	Anzeige
↑	10 PRINT "HELLO
← ← ← ← ←	10 PRINT "HELLO"

Beachten Sie, daß der Cursor die Form eines blinkenden Rechtecks angenommen hat, um so anzuzeigen, daß er auf einem schon bestehenden Zeichen steht. Geben Sie folgendes ein:

Eingabe	Anzeige
GOODBYE"!	10 PRINT "GOODBYE"!_

Erinnern Sie sich, daß Sie einfache numerische Variablen mit Buchstaben von A bis Z belegen.

A=5

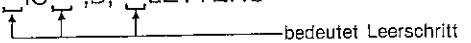
Um Zeichenfolgen-Variablen zu bestimmen, benutzen Sie ebenfalls einen Buchstaben, gefolgt von einem Dollarzeichen. Benutzen Sie niemals den gleichen Buchstaben, um eine numerische und eine Zeichenfolgen-Variable zu bestimmen. Sie können nicht A und A$ im selben Programm benutzen.

Erinnern Sie sich, daß einfache Zeichenfolgen-Variablen nicht mehr als 7 Zeichen umfassen können.

A$="TOTAL"

Die einer Variablen zugewiesenen Werte können sich im Laufe eines Programms ändern, sie können jeweils den im Programm eingegebenen oder errechneten Wert annehmen. Eine Möglichkeit der Variablenzuweisung ist, den Befehl INPUT zu benutzen. Im folgenden Programmbeispiel ändert sich der Wert der Variablen A$ aufgrund der Daten, die eingegeben werden, um die Frage "WORD?" zu beantworten. Geben Sie dieses Programm ein:

```
10: INPUT "WORD?";A$
20: B=LEN (A$)
30: PRINT "WORD␣IS␣";B;"␣LETTERS"
40: END
```



Da Zeile 30 des Programms länger als 24 Zeichen ist, wird der verbleibende Teil in der folgenden Zeile angezeigt.

Das zweite neue Element in diesem Programm ist der Gebrauch der END-Anweisung, um die Beendigung des Programms anzuzeigen. END läßt den Computer wissen, daß das Programm beendet ist. Es ist von Vorteil, beim Programmieren immer eine END-Anweisung zu benutzen.

Wenn Ihre Programme komplexer werden, möchten Sie sie möglicherweise noch einmal durchsehen, bevor Sie sie arbeiten lassen. Zu diesem Zweck benutzen Sie das LIST-Kommando. LIST, das nur im PRO-Modus benutzt werden kann, bringt das gesamte Programm, angefangen bei der niedrigsten Zeilennummer, zur Anzeige.

Versuchen Sie, dieses Programm auflisten zu lassen.

Eingabe	Anzeige
LIST **ENTER**	**10: INPUT "WORD?"; A$** **20: B = LEN (A$)** **30: PRINT "WORD IS ";B;"** **LETTERS"**

Benutzen Sie die ↑- und ↓-Pfeile, um sich in Ihrem Programm zu bewegen, bis Sie das gesamte Programm durchgesehen haben. Um eine Zeile noch einmal anzusehen, die mehr als 24 Zeichen enthält, bringen Sie den Cursor an die äußerste rechte Seite des Displays und die weiteren Zeichen werden in der Anzeige erscheinen. Nachdem Sie Ihr Programm kontrolliert haben, lassen Sie es ablaufen:

Eingabe	Anzeige
CLS RUN **ENTER**	**RUN** **WORD?_**
HELP	**RUN** **WORD? HELP_**
ENTER	**RUN** **WORD? HELP** **WORD IS 4. LETTERS**

Dies ist das Ende Ihres Programms. Natürlich können Sie es noch einmal ablaufen lassen, indem Sie RUN eingeben. Auf jeden Fall wäre das Programm unterhaltsamer, wenn es mehr als eine Eingabemöglichkeit enthielte. Wir wollen das Programm nun so modifizieren, daß es weiterläuft, ohne daß man nach jeder Antwort RUN eingeben muß.

Gehen Sie zurück in den PRO-Modus und benutzen Sie den Auf- oder Abwärts-Pfeil (oder LIST), um in die Zeile 40 zu kommen. Betätigen Sie den Auf- oder Abwärtspfeil, bis sich die Zeile 40 am oberen Rand des Bildschirms befindet (oder LIST 40 eingeben und **ENTER** betätigen).

Eingabe	Anzeige
LIST 40 **ENTER**	**40:END**

Sie können nun 40 eingeben, um so die gesamte Zeile zu löschen. Sie können aber auch den Cursor ▶ auf das E in END setzen. Ändern Sie die Zeile 40 folgendermaßen:

40: GOTO 10

Nun lassen Sie das modifizierte Programm ablaufen.

Die GOTO-Anweisung schafft eine Programm-Schleife (d.h. eine bestimmte Operation wird immer wieder durchgeführt). Da Sie der Schleife kein Ende gesetzt haben, wird sie sich nicht unterbrechen (eine "endlose" Schleife). Um dieses Programm zu unterbrechen, bedienen Sie die BREAK(**BRK**)-Taste.

Wenn Sie ein Programm mit der **BRK**-Taste unterbrochen haben, können Sie es mit dem CONT-Kommando neu starten. CONT steht für CONTinue (fortsetzen). Bei Eingabe des CONT-Kommandos läuft das Programm von da an weiter, wo es mit der **BRK**-Taste unterbrochen wurde.

Beispiel 4 — Komplexere Programmierung

Das folgende Programm berechnet die N-Fakultät (N!). Das Programm beginnt mit 1 und berechnet N! bis zu der Grenze, die Sie eingeben. Geben Sie das Programm ein:

```
100: F=1: WAIT 118
110: INPUT "LIMIT?"; L
120: FOR N=1 TO L
130: F=F*N
140: PRINT N, F
150: NEXT N
160: END
```

In diesem Programm sind einige neue Erscheinungen enthalten. Der Befehl WAIT in Zeile 100 kontrolliert den Zeitraum, über den eine Zeile im Display bleibt, ehe das Programm fortgesetzt wird. Die Zahlen und ihre Fakultäten werden angezeigt, wenn sie berechnet werden. Durch die WAIT Anweisung bleiben sie etwa 2 Sekunden in der Anzeige, statt darauf zu warten, daß Sie **ENTER** drücken.

Ebenfalls in Zeile 100 tauchen zwei Anweisungen auf, die durch einen Doppelpunkt (:) getrennt werden. Sie können so viele Anweisungen in eine Zeile schreiben, wie Sie möchten, solange Sie die einzelnen Anweisungen durch Doppelpunkte voneinander trennen und die maximale Anzahl von 80 Zeichen pro Zeile, einschließlich **ENTER**, nicht überschreiten. Solche Mehrfach-Anweisungen machen allerdings das Lesen und Schreiben eines Programms schwieriger, deshalb ist es besser, sie nur zu benutzen, wenn die Einzel-Anweisungen sehr kurz und einfach sind oder wenn es einen besonderen Grund dafür gibt, Mehrfach-Anweisungen zu verwenden.

In unserem Programm haben wir in Zeile 120 den Befehl FOR und in Zeile 150 den Befehl NEXT benutzt, um eine Schleife aufzubauen. In Beispiel 3 hatten Sie eine "endlose" Schleife geschaffen, die sich kontinuierlich wiederholte. Mit dieser FOR/NEXT-Schleife addiert der PC-1360 jedes Mal 1 zu N hinzu, wenn der Programmablauf den Befehl NEXT erreicht. Hierbei wird geprüft, ob N die anfangs gesetzte Grenze überschreitet. Ist N kleiner oder gleich der Grenze, beginnt der Programmdurchlauf wieder am Anfang der Schleife. Ist N größer als die Grenze, geht der Programmablauf in Zeile 160 weiter und wird dort beendet.

In einer FOR/NEXT-Schleife können Sie jede numerische Variable benutzen. Sie müssen auch nicht mit 1 anfangen, und Sie können bei jedem Programmschritt jeden beliebigen Betrag dazu addieren. Näheres finden Sie in Kapitel 9.

Wir haben dieses Programm mit Zeilennummern von 100 aufwärts belegt. Eine solche Belegung mit unterschiedlichen Zeilennummern macht es möglich, daß man mehrere Programme zur gleichen Zeit im Arbeitsspeicher hat. Um dieses Programm anstelle des Programms der Zeile 10 durchlaufen zu lassen, geben Sie folgendes ein:

```
CLS
RUN 1ØØ
```

Außer der Möglichkeit, Programme durchlaufen zu lassen, indem Sie die erste Zeilennummer eingeben, können Sie den Programmen auch einen Buchstaben als Namen geben und sie mit der DEF-Taste starten (siehe Kapitel 6).

Das Speichern von Programmen im PC-1360

Sie erinnern sich sicherlich, daß Einstellungen, ReSerVe-Tasten und Funktionen im Computer erhalten bleiben, auch wenn Sie ihn abschalten. Programme bleiben ebenfalls im Speicher erhalten, wenn Sie den PC-1360 abschalten oder er das automatisch tut. Selbst, wenn Sie die BRK-, CLEAR- oder CA-Tasten benutzen, bleiben die Programme erhalten.

Programme gehen nur aus dem Speicher verloren, wenn Sie einen der folgenden Abläufe ausführen:

- NEW eingeben, ehe Sie mit dem Programmieren beginnen.
- Den Computer mit dem ALL RESET-Schalter initialisieren.
- Ein neues Programm mit DENSELBEN ZEILENNUMMERN schreiben, die Sie bereits an ein im Speicher vorhandenes gegeben haben.
- Die Batterien wechseln.

Diese kurze Einführung in das Programmieren mit dem PC-1360 sollte dazu dienen, die aufregenden Möglichkeiten des Programmierens mit Ihrem neuen Computer näher zu beschreiben. Für weitere Übungen, siehe Programmbeispiele.

Grafikfunktionen auf dem Bildschirm

Das Anzeigefeld (Bildschirm) des PC-1360 setzt sich aus 150 horizontalen und 32 vertikalen Punkten zusammen. Mit diesen Punkten können einfache Grafiken erstellt werden. Die folgenden 6 Befehle werden dafür bereitgestellt:

GPRINT:	Graphic PRINT. Dieses Kommando erzeugt Formen mit 8 vertikalen Punkten pro Einheit.
GCURSOR:	Graphic CURSOR. Dieses Kommando setzt den Cursor, um mit dem Kommando GPRINT Bilder zu erstellen.
PSET:	Point SET. Dieses Kommando setzt einen bestimmten Punkt an der angegebenen Stelle.
PRESET:	Point RESET. Dieses Kommando löscht einen Punkt an der angegebenen Stelle.
LINE:	Mit diesem Kommando wird zwischen zwei definierten Punkten eine Linie oder ein Quadrat erzeugt.
POINT:	Dieses Funktion ermöglicht die Abfrage, ob ein Punkt gesetzt ist oder nicht.

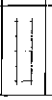
Hier soll nur das Konzept der Erstellung von Grafiken erläutert werden. Genauere Informationen über die Befehle finden Sie unter der entsprechenden Beschreibung des Kommandos.

Es gibt 2 Grundversionen, ein Bild zu erzeugen.

Einmal kann man ein Bild aus vordefinierten Symbolen zusammenstellen, die zweite Möglichkeit ist, das Bild aus einzelnen Punkten zu bilden, die man hell definiert oder dunkel läßt. Um eine Grafik mit Zeichen und alphanumerischen Werten auf dem Bildschirm zu kombinieren, muß das Grafik-Kommando nach der Anzeige der Zeichen und der numerischen Werte gegeben werden.

(1) Bilder mit vordefinierten Symbolen

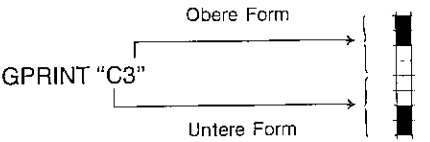
Mit dem GPRINT-Kommando und den 16 vorgesehenen Symbolen, die in der Tabelle angegeben sind, soll ein Bild erstellt werden.

Hexadecimal Zahl	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Punktmuster																

Alle Formen in dieser Tabelle benutzen 4 Punkte.

Im GPRINT-Kommando werden jedoch zwei Formen vertikal zu einer 8-Punkt-Form zusammengesetzt, wie es im unteren Beispiel angegeben ist. Das Bild wird durch Kombinieren und Aneinanderreihen einer Anzahl von 8-Punkt-Formen erstellt.

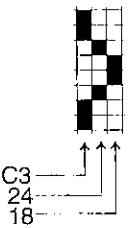
(Beispiel)



Zwei hexadezimale Zahlen werden benutzt, wobei die erste die untere Form und die zweite die obere Form spezifizieren.

(Beispiel)

GPRINT "C32418"

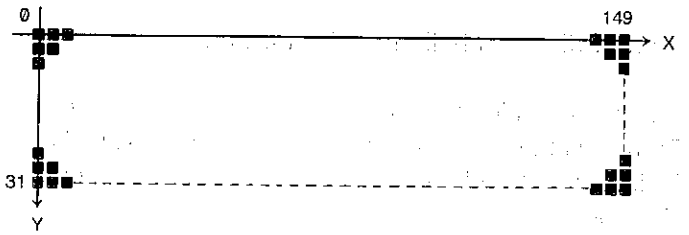


Die durch die Paare von hexadezimalen Zahlen spezifizierten 8-Punkt-Formen ergeben das Bild.

(2) Die Position auf dem Bildschirm

Wie beschrieben können Formen mit Hilfe des GPRINT-Kommandos erstellt werden. Mit dem Kommando GCURSOR kann man die Position des Bildes auf dem Display bestimmen. Das Anzeigefeld (Bildschirm) des PC-1360 wird aus 150 × 32 Punkten gebildet.

Die Position jedes Punktes kann, wie in einem X-Y-Koordinatensystem, mit (x,y) ermittelt werden, wobei "x" die horizontale und "y" die vertikale Achse bilden.



Zu beachten ist, daß ein Punkt umso tiefer auf dem Bildschirm ist, je größer der Wert von "y", also genau entgegen dem normalen X-Y-Koordinatensystem.

Die Koordinaten der Punkte auf dem Bildschirm reichen von 0-149 für "x" und von 0-31 für "y".

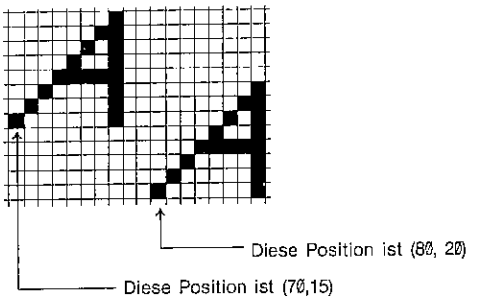
Achtung:
In Wirklichkeit akzeptiert der Computer die Werte für "x" und "y" im Bereich von −32768 und +32767. Wenn ein Wert innerhalb dieses Bereichs, aber außerhalb der Grenzen des Bildschirms liegt, wird ignoriert und hat keinen Einfluß auf den Bildschirm. Dies gilt auch für alle grafischen Bildschirm-Kommandos, einschließlich PSET, PRESET und LINE.

Das GPRINT-Kommando zeichnet Bilder, indem Gruppen von 8-Punkt-Symbolen aneinandergesetzt werden. Wenn die Position (Punkt), an der das Bild erstellt werden soll, mit dem GCURSOR-Kommando angegeben wurde, wird das erste 8-Punkt-Symbol auf den 8 oberen Punkten und einschließlich des Punktes an der angegebenen Position abgebildet. Der Rest des Symbols wird der Reihe nach erstellt.

(Beispiel)

```
 5 CLS
1Ø AA$ = "8Ø4Ø2Ø1Ø181412FF"
2Ø GCURSOR (7Ø, 15)
3Ø GPRINT AA$
4Ø GCURSOR (8Ø, 2Ø)
5Ø GPRINT AA$
```

- Wird das Programm gestartet, wird das folgende Symbol in der Nähe der Bildschirmmitte abgebildet.



(3) Erstellung eines Bildes aus einzelnen Punkten

Ein Bild wird erstellt, indem der vorgegebene Punkt mit Hilfe des Kommandos PSET oder PRESET gesetzt oder ge-löscht wird. Jeder Punkt ermittelt sich auf die gleiche Weise wie im GCURSOR-Kommando.

(Beispiel)

PSET (75,15)	setzt einen Punkt bei (75, 15)
PSET (75,15),X	setzt einen Punkt bei (75, 15), wenn die Position dunkel ist und löscht einen Punkt, wenn er bereits gesetzt war (die Angabe von "X" invertiert den Punkt)
PRESET (75, 15)	löscht den Punkt bei (75, 15).

(Beispiel)

```
100: WAIT 0 : DEGREE
110: FOR A=180 TO -180
     STEP -2
120: B=SIN A*12          } *1
130: C=COS A*12          }
140: X=B+80              } *2
150: Y=C+14              }
160: PSET (X, Y)         ←*3
170: NEXT A
180: WAIT : GPRINT       ←*4
```

*1 Bestimmt die Koordinaten des Umfangs eines Kreises mit dem Radius 12 und dem Mittelpunkt bei (Ø, Ø).
*2 Setzt den Mittelpunkt des Kreises auf (8Ø, 14).
*3 Setzt den Punkt mit den Koordinaten (x, y).
*4 Hält den gezeichneten Kreis.

Das Programm zeichnet einen Kreis mit dem Radius 12 und dem Mittelpunkt bei (8Ø, 14).

Der Winkel wird in Zweierschritten von +180 bis −180 Grad mit einer FOR-NEXT-Schleife geändert. Für jeden Winkel werden die Koordinaten des Umkreises bestimmt und der entsprechende Punkt wird gesetzt.

(4) Linien und Quadrate

Obwohl man Linien und Quadrate auch mit Hilfe des PSET-Kommandos erstellen kann, können sie leicht gezeichnet werden, indem einfach 2 Punkte im LINE-Kommando angegeben werden.

(Beispiel)

```
LINE (Ø, Ø)—(149,31)
  Zieht eine Linie zwischen den Punkten (Ø, Ø) und (149,31).
LINE (3Ø, Ø)—(8Ø,31),B
```

Zeichnet ein Quadrat, dessen Diagonale zwischen den Punkten (3Ø, Ø) und (8Ø,31) liegt. Wird statt "B" "BF" gesetzt, so wird das Quadrat gefüllt.

LINE (3Ø, Ø)—(8Ø, 31), X, BF
Zeichnet ein gefülltes Quadrat. Befindet sich aber in diesem Quadrat ein gesetzter Punkt, so wird er ge-löscht.
(Die Angabe von "X" invertiert die Punkte.)

(Beispiel)

```
2ØØ "A": WAIT Ø
21Ø LINE (6Ø, Ø)—(1ØØ, 31)
    ,X,BF
22Ø GOTO 21Ø
```

Dieses Programm zeichnet ein Quadrat mit der Diagonalen von (6Ø, Ø) bis (1ØØ, 31) und füllt und löscht es. Während das vorher mit den Kommandos GPRINT und PSET erzeugte Bild noch auf dem Bildschirm abgebildet ist, lassen Sie das Programm durch Betätigen von **DEF** und **A** ablaufen. Das Bild im Quadrat wird invertiert.

Achtung:
Der Bereich der grafischen Wiedergabe ist auf der linken Seite 6 Punkte breiter als der normale Wiedergabebereich für alphanumerische Zeichen. Der PC-1360 hält das erstellte Bild auf dem Bildschirm, bis es gelöscht wird. Am Ende eines Programms oder bei einer Unterbrechung kann es also vorkommen, daß auf der linken Seite oder im oberen Bereich das Bild stehen bleibt.
(Das Bild wird erst gelöscht, wenn die Taste **CLS** gedrückt wird.)

Achtung:
Es kann sein, daß Zeichen oder Zahlen, die eingegeben werden, wenn das Programm durch das Grafik-Kommando unterbrochen wird, nicht wiedergegeben werden. Zuerst das Anzeigefeld (Bildschirm) mit Hilfe der Taste **CLS** löschen und dann die Eingabe durchführen.

KAPITEL 6 ABKÜRZUNGEN

Der PC-1360 bietet Ihnen verschiedene Möglichkeiten, die Programmierung zu vereinfachen, indem Sie die Anzahl der Tastenbedienungen und anderer sich wiederholender Abläufe reduzieren.

Eine dieser Möglichkeiten besteht darin, Befehle und Kommandos abzukürzen (siehe Kapitel 9).

In diesem Kapitel sollen vor allem zwei Möglichkeiten erörtert werden, mit deren Hilfe Sie unnötige Tastenbedienung vermeiden können-die Benutzung der DEF-Taste und des Reserve-Modus.

Die DEF-Taste und benannte Programme

Es wird häufiger vorkommen, daß Sie mehrere verschiedene Programme zur gleichen Zeit im Speicher des PC-1360 abspeichern wollen. (Vergessen Sie nicht, verschiedene Zeilennummern zu vergeben.) Um ein Programm mit einem RUN- oder GOTO-Befehl zu starten, müssen Sie sich normalerweise an die erste Zeilennummer eines jeden Programms erinnern (siehe Kapitel 9). Aber es gibt eine einfachere Möglichkeit! Sie können jedes Programm mit einem einzelnen Buchstaben benennen und das Programm dann mit nur zwei Tastenbedienungen mit Hilfe von DEF starten.

Setzen Sie den Kennbuchstaben in die erste Zeile des Programms, auf das Sie sich später beziehen wollen. Der Kennbuchstabe muß in Anführungszeichen eingeschlossen und von einem Doppelpunkt gefolgt sein.

```
10: "A": PRINT "FIRST"
20: END
80: "B": PRINT "SECOND"
90: END
```

Jedes der folgenden Zeichen kann als Kennbuchstaben benutzt werden: A, S, D, F, G, H, J, K, L, = Z, X, C, V, B, N, M und SPC. Wie Sie sehen, sind dies alle Tasten in den letzten beiden Reihen der Tastatur, die durch eine weiße Linie getrennt sind, um Ihnen die Arbeit zu erleichtern.

Um das Programm ablaufen zu lassen, brauchen Sie nun nicht RUN 80 oder GOTO 10 einzugeben, sondern Sie drücken nur die **DEF**-Taste und dann den Kennbuchstaben. Im obigen Beispiel würde, wenn Sie **DEF** und dann "B" drücken würden, "ZWEITES" in der Anzeige erscheinen.

Wenn Sie **DEF** benutzen, um ein Programm zu starten, werden Variablen und Modus-Einstellungen ebenso verarbeitet, als wenn Sie GOTO benutzen würden. Für weitere Einzelheiten, siehe Kapitel 9.

RESERVE-Modus

Ein anderer Weg, mit dem PC-1360 Zeit einzusparen, ist der ReSerVe-Modus.

Innerhalb des Speichers des PC-1360 sind 144 Zeichen für den "Reserve-Speicher" vorgesehen. Sie können diesen Speicherteil benutzen, um häufig gebrauchte Ausdrücke zu speichern. Um diese wieder abzurufen, brauchen Sie nur zwei Tasten zu bedienen.

Versuchen Sie, das folgende Beispiel für das Speichern und Wiederabrufen einer reservierten Zeichenfolge nachzuvollziehen.

Schalten Sie den PC-1360 in den RESERVE-Modus um, indem Sie die **SHIFT**- und **MODE**-Tasten drücken. Beachten Sie, daß die Modus-Anzeige "RUN" und "PRO" verschwindet und die Nachricht "RESERVE MODE" auf dem Bildschirm erscheint, wenn Sie den RESERVE-Modus eingeschaltet haben.

Geben Sie NEW ein und drücken dann die **ENTER**-Taste. Dadurch löschen Sie alle vorher gespeicherten Daten, so, wie Sie mit NEW im PROgramm-Modus alle vorher gespeicherten Programme löschen.

Geben Sie **SHIFT** gefolgt von 'A' ein:

Eingabe	Anzeige
SHIFT A	A:_

Beachten Sie, daß 'A' links in der Anzeige erscheint und von einem Doppelpunkt gefolgt wird.

Geben Sie das Wort 'PRINT' ein und betätigen Sie die **ENTER**-Taste.

Eingabe	Anzeige
PRINT **ENTER**	**A: PRINT**

Ein Leerschritt erscheint nach dem Doppelpunkt und zeigt an, daß "PRINT" nun im Reservespeicher unter dem Buchstaben "A" gespeichert worden ist.

Schalten Sie den PC-1360 in den PROgramm-Modus um. Geben Sie NEW, gefolgt von **ENTER** ein, um den Programmspeicher zu löschen. Geben Sie '10' als Zeilennummer ein und betätigen Sie dann die **SHIFT**-Taste und die 'A'-Taste:

Eingabe	Anzeige
1Ø **SHIFT**	**10 PRINT _**
ENTER	**10: PRINT**

Sofort wird das Wort 'PRINT' hinter der Zeilennummer in der Anzeige erscheinen.

Im Reserve-Speicher kann jede beliebige Zeichenfolge gespeichert werden. Die gespeicherten Zeichenfolgen können jederzeit im PROgramm- oder RUN-Modus wieder abgerufen werden, indem Sie **SHIFT** drücken und anschließend den Buchstaben, unter dem Sie die Zeichenfolge abgespeichert haben. Die dafür zur Verfügung stehenden Tasten sind dieselben wie im Zusammenhang mit DEF erwähnt, also alle Tasten, die von der weißen Linie umgerahmt sind.

Die maximale Länge einer Reservezeichenfolge für eine Taste ist 80 Tastenanschläge, einschließlich des Tastennamens und der **ENTER**-Taste, um die Definition zu beenden. Alle BASIC-Kommandos, -Befehle oder -Funktionen benutzen je zwei Byte der 144 Byte, die im Reservespeicher verfügbar sind.

Um eine gespeicherte Zeichensequenz aufzubereiten, schalten Sie den Reserve-Modus ein und drücken **SHIFT**, gefolgt von dem Buchstaben, unter dem die Sequenz gespeichert ist. Mit Hilfe der Links- und Rechts-Pfeile, der DEL- und der INS-Tasten können Sie nun genauso aufbereiten wie in den anderen Modi.

Ist das letzte Zeichen in einer abgespeicherten Sequenz das "GOTO 100@"-Zeichen, wird es bei Wiederabruf der Sequenz als **ENTER** ausgegeben. Speichern Sie beispielsweise die Zeichenfolge "GOTO 100" unter der "G"-Taste, so wird bei Betätigung von **SHIFT** und "G" im RUN-Modus augenblicklich die Ausführung des Programms in Zeile 100 beginnen. Ohne das "@"-Zeichen müssen Sie nach dr **SHIFT** "G"-Sequenz **ENTER** eingeben, um das Programm zu starten.

Um Reserveprogramme zu löschen:

1. Wie Sie wissen, löscht die Tastenfolge NEW und **ENTER** alle Reserve-Speicher.
   Denken Sie daran, daß die oben beschriebene Operation im RESERVE-Modus durchgeführt werden muß.

2. Um einen Reserve-Speicher zu löschen, benutzen Sie die **SPC**- oder **DEL**-Taste, wie unten beschrieben:
   Beispiel: Löschen Sie A*A, was unter der Zeilennummer **S** gespeichert ist.

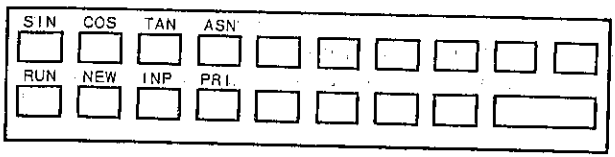
Eingabe	Anzeige	Bemerkungen
SHIFT S	S:_	Reserve-Modus
A*A **ENTER**	S:A*A	
CLS	>	
SHIFT S	S:A*A	
◄ oder ►	S A*A	
DEL **DEL** **DEL**	S:_	Löscht A*A
ENTER	>	

Schablonen

Zum PC-1360 gehört eine Schablonen. Sie können diese benutzen, um das Erinnern an häufig gebrauchte Reserve-Sequenzen oder Kennbuchstaben-Zuweisungen über DEF zu erleichtern. Nachdem Sie den Programmen Kennbuchstaben zugewiesen haben oder Sequenzen eingerichtet haben, markieren Sie die Schablone, damit Sie wissen, welche Kennung zu welcher Taste gehört. Sie können dann mit nur zwei Tastenbetätigungen Programme abarbeiten lassen oder Sequenzen abrufen.

Wenn Sie beispielsweise eine Reihe von Programmen haben, die Sie häufig zur gleichen Zeit benutzen, vergeben Sie Kennbuchstaben und markieren die Schablone, so daß Sie die Programmausführung ganz einfach mit nur zwei Tastenbedienungen starten können. Sie können auch häufig gebrauchte BASIC-Befehle und Kommandos in den Reserve-Speicher einspeisen und eine Schablone entsprechend markieren, um so die Eingabe von BASIC-Programmen zu beschleunigen.

Beispiel:



KAPITEL 7 ZUSÄTZLICHE PERIPHERIGERÄTE

Der PC-1360 arbeitet mit den folgenden zusätzlichen SHARP Peripheriegeräten:

MODELL	BESCHREIBUNG	VERBINDUNGSSTECKER
CE-126	Einfacher Thermaldrucker	Druckerinterface
CE-140P	7-Farben-Matrixdrucker	Serielles E/A-Interface
CE-515P/516P	4-Farben-Plotter/Drucker	Serielles E/A-Interface
CE-124	Kassetteninterface	Druckerinterface
CE-152	Kassettenrecorder	Über CE-126P oder CE-124
CE-130T	Pegelumsetzer	Serielles E/A-Interface

Jedes Gerät wird hier unten kurz beschrieben. Für weitere Einzelheiten, lesen Sie bitte in den entsprechenden Bedienungsanleitungen nach. Lesen Sie das Kapitel über die RAM-Karten für weitere Einzelheiten über den Gebrauch der RAM-Karten mit dem PC-1360.

CE-126P Thermaldrucker/Kassetteninterface

Mit diesem optionellen CE-126P Drucker/Kasetten-Interface können Sie einen Drucker und einen Kassettenrecorder and Ihren SHARP PC-1360-Computer anschließen.
Der CE-126P besitzt die folgenden Besonderheiten:

- Thermaldrucker mit einer Druckbreite von 24 Zeichen.
- Praktischer Papiervorschub und Abrißkante
- Gleichzeitiges Drucken der Berechnungen nach Wunsch.
- Leichte Regelung des Displays bzw. der Druckerausgabe in BASIC.
- Eingebautes Kassetteninterface mit Fernfunktion.
- Manuelle und programmierte Steuerung des Rekorders zum Abspeichern von Programmen, Daten.
- Trockenbatteriebetrieb zwecks Portabilität.

Über den Anschluß des PC-1360 an den CE-126P, beziehen Sie sich bitte auf die dem CE-126P mitgelieferten Bedienungsanweisungen.

CE-140P Mehrfarben Matrixdrucker

Der CE-140P ist ein Mehrfarben-Matrixdrucker, der wiederaufgeladen werden kann. Körperlich ist er viel größer als der CE-126P und hat eine Vertiefung, in die der Computer passt, um eine integrale Einheit zu bilden. Eine Vierfarben-Patrone erlaubt das Drucken in 7 Farben. Die Papierbreite beträgt 114 mm mit bis zu 80 Zeichen pro Zeile. Der WS-Adapter EA-150 dient zum Aufladen des Gerätes. Zu den Charakteristika gehören ein Text-Modus zum Ausdrucken von Zeichen, Zahlen und Symbolen verschiedener Größe, und ein grafischer Modus zum Zeichnen von Figuren. Die Position und die Richtung des Druckkopfes können eingestellt und kontinuierliche gestrichelten Linien verschiedener Form können gezeichnet werden. Der Drucker benutzt auch ein Schreibfenster, um das Drucken außerhalb der Seite zu ignorieren, ohne Fehler zu erzeugen.

Der PC-1360 hat eine Anzahl von speziellen grafischen Kommandos, die für den Gebrauch mit dem CE-140P gedacht sind (siehe das Kapitel über die Computer-Anweisungen des PC-1360).

Obwohl der CE-140P an das serielle E/A-Interface angeschlossen ist, erkennt ihn der PC-1360 als Drucker und behandelt ihn wie den an das Druckerinterface angeschlossen CE-126P. Das Drucken und Auflisten erfordern kein Öffnen des seriellen E/A-Kanals (siehe LLIST und LPRINT-Befehle).

CE-515P/516P Farb-Plotter/Drucker

Dieses Gerät ist ein Vierfarben-Plotter/Drucker. Anders als der CE-140P Matrixdrucker, zeichnet er mit Hilfe von Tintenstiften und ist sehr geeignet zum Zeichnen von grafischen Darstellungen und Diagrammen von bester Qualität. Es ist ein betriebssystemunabhängiges Gerät, das über ein Kabel an das serielle E/A-Interface des Computers angeschlossen wird.

Der Plotter/Drucker hat viele der Charakteristika des CE-140P Matrixdruckers einschließlich eines Text-Modus für Zeichen, Zahlen und Symbole verschiedener Größe und eines grafischen Modus zum Plotten und Zeichnen. Ein Zeichenfenster ignoriert das Zeichnen außerhalb der Seite ohne Fehler zu erzeugen. Sowohl absolute wie auch relative Stiftbewegungen sind möglich. Der CE-516P ist eine verbesserte Version des CE-515P, mit erweiterten Zeichen- und Symbolgruppen, 63 Zeichengrößen anstelle von 15 und solchen Zeichenfunktionen wie Bogen, Kreis, Kasten, Ellipse und feste Schraffur.

Der PC-1360 hat eine Anzahl von speziellen grafischen Kommandos, die für den Gebrauch mit dem CE-515P/516P gedacht sind (siehe das Kapitel über die Computer-Anweisungen des PC-1360).

CE-130T Pegelumsetzer

Der CE-130T ist ein kleiner Adapter, der zur Regulierung der Eingangs- und Ausgangspegel des seriellen E/A-Interface eingesetzt wird, um den Normen des RS-232C Interface zu genügen. Der Zweck dieses Gerätes ist es, die Verknüpfung des PC-1360 mit einer möglichst großen Auswahl von anderen RS-232C-kompatiblen Peripheriegeräten und Computern zu ermöglichen. Z.B. kann eine Verknüpfung von Computer zu Computer für die direkte Übertragung von Programmen benutzt werden. Das Gerät hat eine Ni-Cd-Batterie, die mit Hilfe des WS-Adapters EA-11E wiederaufgeladen werden kann.

CE-124 Kassetteninterface

Dieses Gerät wird benutzt, um den Computer an einen Kassettenrecorder zum Abspeichern und Laden von Programmen auf Band anzuschließen (sofern nicht das eingebaute Kassetten-Interface auf dem CE-126P Drucker eingesetzt wird). Es wird in das Druckerinterface auf der Computerseite und in die MICrophone- und Kopfhörer- Buchsen des Recorders angeschlossen. Wir empfehlen den Gebrauch des Kassettenrecorders SHARP CE-152, obwohl Sie auch andere verwenden können, sofern diese die folgenden Spezifikationen erfüllen:

Posten	Anforderungen
1. Recordertyp	Jedes Kassettengerät oder Mikrokassettengerät
2. Aufnahmeanschluß	Mini-Klinkensteckeranschluß mit der Bezeichnung "MIC" (niemals "AUX"-Anschluß)
3. Aufnahmeimpedanz	Anschluß mit niedriger Impedanz (200—1000 Ohm)
4. Minimaler Aufnahmepegel	Unter 3 mV oder −50 dB.
5. Wiedergabeanschluß	Mini-Klinkensteckeranschluß mit der Bezeichnung "EXT" (EXTerner Lautsprecher), "MONITOR", "EAR" (Kopfhörer) oder ähnlichem
6. Wiedergabeimpedanz	Unter 10 Ohm
7. Wiedergabepegel	Über 1V (maximale Wiedergabe über 100 mW)
8. Verzerrung	Innerhalb 15% (Bereich: 2 bis 4 KHz)
9. Ieichlaufschwankung	Maximal 0,3% (W.R.M.S.)
10. Anderes	Motor des Recorders ohne Geschwindigkeitsschwankungen

Verwendung des CE-126P Drucker/Kassetten-Interface

Verwendung des Druckers

Wenn Sie den PC-1360 für normale Berechnungen verwenden, können Sie den CE-126P zum simultanen Ausdruck Ihrer Berechnungen verwenden.

Dazu ist es nur notwendig, im RUN-Modus die Tasten **SHIFT** und dann **ENTER** (P⟷NP) zu drücken.

Wenn Sie dann nach Abschluß der Berechnung die Taste **ENTER** drücken, wird der Inhalt der Anzeige in einer Zeile und das Ergebnis der Rechnung auf der nächsten Zeile ausgedruckt. Zum Beispiel:

Eingabe	Papier
300/50 **ENTER**	300/50 6.

Sie können auch von BASIC-Programmen aus mit der LPRINT-Anweisung Ausdrucke veranlassen (Einzelheiten dazu in Kapitel 9). LPRINT kann wie die PRINT-Anweisung verwendet werden.

Programme, die die PRINT-Anweisung verwenden, können durch die Eingabe der Anweisung PRINT = LPRINT in das Programm leicht für das Arbeiten mit einem Drucker konvertiert werden (Einzelheiten in Kapitel 9). Alle nachfolgenden PRINT-Anweisungen werden sich wie LPRINT-Anweisungen verhalten. PRINT = PRINT setzt wieder den ursprünglichen Zustand her. diese Konstruktion kann auch in eine IF-Anweisung eingebaut werden, wodurch Sie bei Verwendung des Programmes die Wahl der Art der Ausgabe haben.

Mit der LLIST-Anweisung haben Sie ferner die Möglichkeit, Ihr Programm von Drucker ausdrucken zu lassen (Einzelheiten in Kapitel 9). Eine LLIST-Anweisung ohne Zeilennummern veranlasst den Ausdruck des gesamten Programmes. Durch Angabe von Zeilennummern mit der LLIST-Anweisung wird nur der Ausdruck der angegebenen Programmteile veranlasst. Wenn Programmzeilen länger als 24 Zeichen sind, werden zwei oder mehr Zeilen zum Ausdruck der Programmzeile verwendet. Die zweite und die nachfolgenden Zeilen werden um vier oder sechs Zeichen eingerückt, um so die unterscheidung der Programmzeilen zu erleicherten. (Zeilenummer 1 bis 999: vier; über 999: sechs.)

ACHTUNG:

- **Im Falle eines Fehlers (ERROR-Code 8) wegen fehlerhafter Papierzuführung reißen Sie das Papierband ab und ziehen Sie den restlichen Teil des Papiers völlig aus dem Drucker. Drücken Sie anschließend die CLS-Taste, um den Fehlerzustand zu löschen.**
- **Wenn der Drucker starken äußeren elektrischen Störungen ausgesetzt ist, kann es zum zufälligen Ausdruck von Zahlen kommen. In diesem Fall**

drücken Sie die BRK-Taste,um das Drucken zu unterbrechen. Schalten Sie den CE-126P aus und wieder ein und drücken Sie danach die CLS-Taste. Das Drücken der CLS Taste setzt den Drucker wieder in den normalen Zustand.

- **Wenn es zu einem Fehler in der Papierzuführung kommt oder der Drucker starken äußeren elektrischen Störungen ausgesetzt ist, arbeitet der Drucker nicht normal. In diesem Fall drücken Sie die BRK-Taste, um das Drucker zu unterbrechen. (Entfernen Sie das fehlerhaft zugeführte Papier.) Schalten Sie den Drucker aus und wieder ein und drücken Sie die CLS-Taste.**
- **Wenn Sie den CE-126P nicht benutzen, schalten Sie ihn aus, um die Batterien zu schonen.**

Bedienung des Kassetten-Interface

Aufnehmen (Speichern) auf Magnetband mit Hilfe des eingebauten Kassetteninterface CE-126P

Siehe auch Anmerkungen zum Tonband.

1. Schalten Sie den REMOTE-Schalter des CE-126P auf OFF.

2. Geben Sie ein Programm oder Daten in den Computer ein.

3. Legen Sie die Kassette in den Recorder ein. Spulen Sie das Band bis zur Stelle vor, an der Sie das Programm abspeichern wollen, wobei Sie darauf achten müssen, daß Sie den Leervorspann (nicht-magnetisierte Folie) und bereits aufgenommene Programme nicht bespielen.

4. Stecken Sie den roten Stecker des Interface in die Mikrofonbuchse MIC des Kassettenrecorders und den schwarzen Stecker in die REM-Buchse.

5. Schalten Sie den REMOTE-Schalter ein.

6. Drücken Sie gleichzeitig die Aufnahme- und Wiedergabe-Tasten des Kassettenrecorders (um diesen in den Aufnahmemodus zu setzen).

7. Geben Sie die Aufnahmeanweisungen ein (CSAVE-Anweisung, PRINT#-Anweisung), und drücken Sie die ENTER-Taste.

Setzen Sie die Einheit zunächst in den RUN- oder PRO-Modus. Drücken Sie dann die folgenden Tasten:
C S A V E SHIFT " dateiname **SHIFT " ENTER.**
(Um den Inhalt des Datenspeichers auf Band zu überspielen, drücken Sie die folgenden Tasten: **P R I N T SHIFT #** variable **ENTER**.)
z.B. **C S A V E SHIFT " A A SHIFT " ENTER**

Wenn Sie die **ENTER**-Taste drücken, wird das Band zu laufen beginnen, wobei ein etwa 8 Sekunden langes Stück unbespielt bleibt. (Ein Piepton wird aufgenommen.) Anschliessend werden der Dateiname und sein Inhalt aufgenommen.

8. Wenn die Aufnahme vollständig ist, erscheint das Bereitschaftssymbol in der Anzeige (>) und der Recorder hält automatisch an. Nun haben Sie Ihr Programm im Kassettenrecorder abgespeichert (es befindet sich jedoch trotzdem im PC-1360).

   Wenn Daten über einen Programmbefehl automatisch abgespeichert werden sollen (PRINT #-Anweisung, nicht bei manueller Bedienung), befolgen Sie erst die Schritte 1 bis 6, ehe Sie das Programm ablaufen lassen.

   Als Hilfe beim Auffinden bestimmter Programme auf Band benutzen Sie bitte das Bandzählwerk des Recorders.

Überprüfung eines abgespeicherten Programms

Nachdem Sie ein Programm auf Band geladen bzw. von Band abgespeichert haben, können Sie überprüfen, ob das Programm im PC-1360 identisch mit dem auf Band ist (und so sicher gehen, daß alles in Ordnung ist, ehe Sie mit dem programmieren oder der Programmdurchführung fortfahren).

1. Schalten Sie den REMOTE-Schalter auf OFF.

2. Bewegen Sie das Band bis zu einer Stelle kurz vor Beginn der zu prüfenden Datei.

3. Stecken Sie den grauen Stecker in die Kopfhörerbuchse und den schwarzen Stecker in die REMote-Buchse.

4. Schalten Sie den REMOTE-Schalter auf ON.

5. Drücken Sie die PLAY-Taste.

6. Geben Sie eine CLOAD?-Anweisung ein und beginnen Sie den Prüfvorgang mit der ENTER-Taste. Gehen Sie dabei folgendermaßen vor: Schalten Sie den RUN- oder PRO-Modus ein. Geben Sie die folgende Tastenfolge ein—

Der Dateiname, den Sie vorher gewählt haben.

C L O A D SHIFT ? SHIFT " A A SHIFT " ENTER

Der PC-1360 wird nun automatisch nach dem vorgegebenen Dateinamen suchen und wird den Inhalt des Tonbandes mit dem Speicherinhalt vergleichen.

Wenn die vorgegebene Datei auf dem Band gefunden ist, erscheint automatisch ein Sternzeichen "*" am Ende der Zeile, die auf dem Bildschirm eingegeben wurde. Dies bedeutet, daß die eigentliche Überprüfung begonnen hat. Werden die Programme als übereinstimmend erkannt, erscheint das Bereitschaftssymbol (>) in der Anzeige.

Decken sich die Programme nicht, wird der Vergleich unterbrochen und die Meldung ERROR 8 (Error code 8) ausgegeben. In diesem Fall, versuchen Sie es bitte noch einmal.

Einladen vom Magnetband

Siehe auch Anmerkungen zum Tonband.

Um Programme oder Daten vom Magnetband in den PC-1360 einzuladen, zu übertragen oder abzulesen, gehen Sie bitte folgendermaßen vor:

1. Schalten Sie den REMOTE-Schalter auf OFF.

2. Legen Sie das Band in den Recorder ein. Bringen Sie das Band kurz vor die Stelle, an der das abzulesende Programm beginnt.

3. Stecken Sie den grauen Stecker in die Kopfhörerbuchse und den schwarzen Stecker in die REM-Buchse. (Falls der Recorder keinen REM-Eingang hat, benutzen Sie die PAUSE-Taste, um den Bandvorlauf manuell auszuführen.)

4. Schalten Sie den REMOTE-Schalter wieder auf ON.

5. Drücken Sie die PLAY-Taste des Recorders (Wiedergabe-Modus).

   Stellen Sie den VOLUME-Regler auf mittlere oder maximale Lautstärke ein.
   Drehen Sie die Höhen maximal auf.

6. Geben Sie die Übertragungsanweisungen (CLOAD-Anweisung, INPUT#-Anweisung) ein und starten Sie die Ausführung mit Hilfe der **ENTER**-Taste.

   Setzen Sie die Einheit in den RUN- oder PRO-Modus. Danach betätigen Sie die folgenden Tasten:

   C L O A D SHIFT " dateiname **SHIFT " ENTER**.
   (Um den Inhalt des Datenspeichers zu laden, drücken Sie: **I N P U T SHIFT #** variable **ENTER**.)

   z.B. **C L O A D SHIFT " A A SHIFT " ENTER**
   Der vorgegebene Dateiname wird nun automatisch gesucht und sein Inhalt wird in den PC-1360 übertragen.

   Wenn die vorgegebene Datei auf dem Band gefunden ist, beginnt das Laden. Dies wird durch ein Sternzeichen "*" angegeben, das am Ende der eingegeben Zeile erscheint.

7. Wenn das Programm übertragen worden ist, hält der Computer automatisch den Bandlauf an und das Bereitschaftssymbol (>) erscheint in der Anzeige.

Um im Verlauf eines Programmes Daten zu übertragen (INPUT#-Anweisung), führen Sie erst die Schritte 1 bis 5 aus, ehe Sie das Programm starten.

Wenn ein Fehlercode 8 erzeugt wird, versuchen Sie noch einmal das Programm von vorn an zu laden. Tritt der Fehler wieder auf, wiederholen Sie die Prozedur nachdem Sie die Lautstärke ein wenig nach oben oder unten eingestellt haben. Wird kein Fehlermeldung ausgegeben und läuft das Band trotzdem weiter, stimmt etwas nicht. Drücken Sie dann die **BRK**-Taste, um das Band zu stoppen und versuchen Sie es noch einmal.

Anmerkungen zum Tonband

1) Benutzen Sie zum Vergleich oder zum Laden immer den Recorder, mit dem das Programm gespeichert wurde. Die Verwendung eines anderen Gerätes kann möglicherweise Fehler erzeugen.

2) Benutzen Sie nur Tonbandkassetten der höchsten Qualität. Benutzen Sie keine Standard-Tonbänder.

3) Halten Sie die Tonköpfe und das Kassettenlaufwerk sauber. Falls beim Lesen, Überprüfen oder Laden von Programmen Fehler auftreten, kann der Grund ein verschmutzter Tonkopf. Reinigen und entmagnetisieren Sie den Tonkopf und wiederholen Sie dann die Prozedur.

4) Stellen Sie die Lautstärke zwischen den Mittelwert und dem Maximalwert ein. Eine falsche Einstellung kann manchmal auch Fehler zur Folge haben. Experimentieren Sie ein wenig mit der Einstellung, bis keine Fehler mehr erzeugt werden.

5) Achten Sie darauf, daß alle Drähte und Verbindungen sauber und richtig angeschlossen sind.

6) Manchmal wird bei die Benutzung des WS-Adapters für das Interface CE-126P ein Summen erzeugt, das das Aufnahmesignal beeinflußt. In diesem Fall, schalten Sie auf Batteriebetrieb um.

7) Stellen Sie den Höhenregler aux maximale Höhen ein.

8) Wenn Sie Programme oder Daten auf einem bereits gebrauchten Band abspeichern wollen, löschen Sie die alten Programme auf dem Band, bevor Sie wieder aufnehmen.

Gebrauch von Farbdruckern

Der Anschluß des zusätzlichen CE-140P Farb-Matrixdrucker bzw. des CE-515P Farb-Plotter/Drucker an den **Computer** ermöglicht es Ihnen, Programme, Rechenergebnisse und Grafiken farbig ausdrucken zu lassen. Der CE-140P kann unter Verwendung einer Fünffarben-Patrone Darstellungen in sieben Farben ausdrucken, während der CE-515P eine grafische Darstellung oder ein Diagramm mit Hilfe von vier verschiedenfarbigen Tintenstiften zeichnet. Der CE-140P kann direkt an das serielle E/A-Interface des Computers angeschlossen werden. Für den Anschluß des CE-515P an das serielle E/A-Interface ist jedoch ein Kabel (CE-516L) erforderlich.

Anmerkungen zu Farbdruckern

Die im **Computer** verwendeten BASIC-Befehle schließen verschiedene Drucker-Befehle ein, die auch für Farb-Matrixdrucker oder Farb-Plotter/Drucker verwendet werden können, wie auch die LLIST und LPRINT Befehle für den Ausdruck von Programmen und Rechenergebnissen.

Der CE-515P Farb-Plotter/Drucker kann nur dann verwendet werden, wenn das serielle E/A-Interface geöffnet ist. Führen Sie also immer den OPEN-Befehl aus, ehe Sie irgendeinen Befehl an den Plotter/Drucker übermitteln.

Darstellungsbereich und Grafik-Koordinaten

Wenn Sie mit dem CE-140P oder dem CE-515P Daten in grafischer Form ausdrucken, können Sie die sogenannte Schneide-Funktion benutzen. Dabei wird der Teil der Grafik, welcher auf dem Papier ausgedruckt wird, tatsächlich dargestellt, während der Teil, der sich außerhalb des Papiers befindet, imaginär dargestellt wird. Diese Funktion ist besonders dann nützlich, wenn Sie eine Grafik nur teilweise zeichnen wollen, es aber einfacher für Sie ist, ein Programm für das Zeichnen der gesamten Grafik zu erstellen, oder wenn die Grafik derart groß ist, daß Sie sie der Papiergröße entsprechend unterteilen müssen.

Ist das Programm allerdings fehlerhaft, mag die Grafik nicht auf das Papier sondern in den imaginären Bereich gezeichnet werden. Achten Sie also bei Verwendung dieser Funktion auf ein fehlerfreies Programm.

Die für das Zeichnen einer Linie beim grafischen Drucken erforderlichen Angaben über Richtung und Position werden durch die Koordinaten X und Y ausgedrückt. Bei Koordinate X (horizontale Richtung) bedeutet "–" nach links und "+" nach rechts. Bei Koordinate Y (vertikale Richtung) bedeutet "+" nach oben und "–" nach unten.

Bei Verwendung eines zusätzlichen Farbdruckers, entnehmen Sie bitte genauere Informationen aus dessen mitgelieferter Bedienungsanleitung.

Serielle E/A-Funktion

Der **Computer** ist mit einem seriellen E/A Interface ausgestattet. Diese Interface-Funktion kann für den Anschluß der zusätzlichen CE-140P bzw. CE-515P Farbdrucker an den **Computer** zum Ausdruck von Grafiken in verschiedenen Farben oder zum Datenaustausch zwischen dem **Computer** und einem anderen Computer verwendet werden.

Beachte: Denken Sie daran, daß das Anlegen einer höheren als der für den **Computer** zulässigen Spannung an das E/A-Interface interne Teile zerstören kann.

Grundlegendes zur Verwendung des seriellen E/A-Interface
Normalerweise ist die Schaltung des seriellen I/O- Interfaces geschlossen. In diesem Zustand können weder Daten empfangen noch Daten gesendet werden. Deshalb müssen Sie die Schaltung mit einem OPEN-Befehl öffnen. (Falls sie schon geöffnet ist, wird der ERROR 8 gemeldet.)
Außerdem ist es nötig, daß den Bedingungen für den Austausch von Daten zwischen dem **Computer** und einem anderen angeschlossenen Computer entsprochen wird. Anders gesagt, die Signal-Bedingungen für den PC-1360 und den angeschlossen Computer müssen die gleichen sein. Sind die Bedingungen verschieden, können die Signale (Daten) nicht korrekt gelesen werden, was Datenfehler zur Folge hat. Der OPEN-Befehl kann zur Festlegung undÄnderung der E/A-Bedingungen verwendet werden.

Nachdem die Bedingungen für beide Seiten erfüllt sind und die Schaltung offen ist, können mit den folgenden Befehlen Daten oder Programme ausgetauscht werden:

LPRINT, LLIST, SAVE, LOAD, PRINT#1, INPUT#1

Nach dem Ende des Austausches von Daten (oder Programmen) wird die Schaltung des seriellen E/A-Interfaces wieder geschlossen. Obwohl der CLOSE-Befehl zur Schließung der Schaltung dient, schließt sich die Schaltung auch, wenn das Programm beendet ist (so wie bei Durchführung des END-Befehles) oder wenn der RUN-Befehl gegeben wird.
Beim Schreiben eines Programmes, welches das serielle E/A-Interface benutzt, muß die Schaltung also zuerst geöffnet werden, dann können Daten ausgetauscht werden und danach sollte die Schaltung, wie oben beschrieben, wieder geschlossen werden.

Beachte: Der **Computer** besitzt keine Timer-Funktion, um den Datenaustausch mit einem angeschlossenen Peripheriegerät durch Messen der Wartezeit für jeden E/A-Befehl zum seriellen E/A Interface zu unterbrechen.
Ist das angeschlossene Gerät während der Ausführung eines Befehles nicht zum Datenaustausch bereit (z.B. nicht eingeschaltet) oder die Verbindung unterbrochen, kann der **Computer** die Befehlsausführung nicht beenden.
Drücken Sie in diesem Fall die BRK Taste und beenden Sie den Programmablauf.

Serielle E/A-Funktion während Verwendung des CE-140P
Der Anschluß des zusätzlichen CE-140P Farb-Matrixdruckers and den **Computer** ermöglicht sofortigen Daten-Ausdruck mit einer einzelnen Kompakt-Einheit.
Der CE-140P besitzt ein serielles E/A-Interface, womit die Verbindung des Druckers und des Computers zu einem anderen Personal-Computer möglich gemacht wird.

- Der CE-140P Schalter sollte normalerweise auf SIO gestellt sein. Der Drucker kann, abhängig von der System-Konfiguartion, wie folgt verwendet werden:

CE-140P Schalter-Stellung	Serielles E/A-Interface	Nur CE-140P angeschlossen	CE-140P und CE-126P angeschlossen*1
SIO (Für normale Verwendung)	OFFEN	E/A vom E/A-Interface ausgeführt	Wie links
	GESCHLOSSEN	Ausdruck über CE-140P	Ausdruck über CE-126P*3
DRUCKER	OFFEN*2	Ausdruck über CE-140P	Wie links
	GESCHLOSSEN	Ausdruck über CE-140P	Ausdruck über CE-126P*3

*1 In diesem Fall ist der CE-126P Drucker zusammen mit dem CE-140P Drucker an der linken seite des **Computers** angeschlossen.

*2 Wenn Sie den CE-140P in dieser Einstellung benutzen, setzten Sie die seriellen E/A-Bedingungen mit dem OPEN-Befehl wie nachfolgend beschrieben fest (Initialisiert: siehe OPEN-Befehl).
OPEN "1200, N, 8, 1, A, C, &1A"
Wenn die Einstellung der E/A-Bedingungen nicht korrekt ist, wird der CE-140P Drucker nicht fehlerfrei arbeiten.

*3 Befehle in Zusammenhang mit Grafik werden an den CE-140P ausgegeben.

Beachte: 1. Ehe Sie den CE-140P an den Computer anschließen, müssen Sie den Computer unbedingt ausschalten.
2. Während die LOW BATTERY-Anzeige am CE-140P leuchtet, wird jeder Versuch, durch Befehle vom Computer aus Daten über den CE-140P ausgeben zu lassen in einem ERROR enden (Errorcode 8).
3. Wenn sich der CE-140P Tintenaustausch-Zustand befindet, kann der Computer nicht abgeschaltet werden, selbst wenn Sie den Netzschalter in die OFF-Stellung schieben. Drücken Sie die INK Taste des CE-140P und betätigen Sie anschließend den Netzschalter des Computers.

Wenn Sie die seriellen E/A-Interface Befehle gebrauchen

Sind in den PC-1360 mehrere Programme mit dem MERGE-Befehl geladen worden, so können diese nicht mit den SAVE-oder LOAD-Befehl über das serielle Interface gesendet werden. Dies deshalb, weil sich der MERGE-Befehl des PC-1360 vom MERGE-Befehl anderer Personal-Computer unterscheidet.

Beachte: 1. Das Anlegen von Spannungen an die Anschlußstifte des seriellen E/A-Interfaces, welche die festgesetzten Grenzwerte überschreiten, kann zur Zerstörung innerer Teile führen.

2. Das serielle Interface des PC-1360 kann nicht unterscheiden, ob der andere Computer bereit ist, Daten zu empfangen oder nicht. Der PC-1360 fährt mit der Ausführung der E/A-Befehle fort und kann von der emfangenden Seite aus nicht gestoppt werden. Wenn der andere Computer aufhört, Daten zu empfangen, betätigen Sie die **BRK** -Taste des PC-1360, um die Datenübertragung zu stoppen.

KAPITEL 8 GEBRAUCH DER RAM-KARTEN

Der PC-1360 hat auf seiner Unterseite zwei Einbauplätze, in die verschiedene RAM-Karten eingelegt werden können. Programme und Daten werden in den Karten erhalten je nach der Speicherstruktur, die mit Hilfe des Kommandos SET MEM gewählt wird. Der PC-1360 selbst besitzt kein internes RAM und Kann daher nicht als Computer bedient werden, wenn nicht wenigstens eine RAM-Karte im Einbauplatz 1 installiert ist. RAM-Karten haben eingebaute Batterien, die den Speicher bei der Herausnahme aus dem Computer schützen. Somit können Programme gespeichert und sichergestellt werden als Alternative zur Speicherung auf Band. Die RAM-Speicherkapazität reicht von 2 Kilobyte bis 32 Kilobyte.

RAM-Kartenname	Kapazität
CE-210M	2 KB
CE-211M	4 KB
CE-212M	8 KB (Standard-Karte)
CE-2H16M	16 KB
CE-2H32M	32 KB

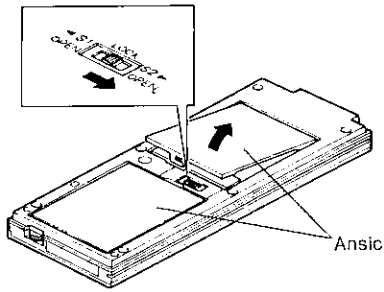
1. Installation einer RAM-Karte

Installieren Sie immer nur eine RAM-Karte auf einmal. Die Arretierung auf der Unterseite des Gerätes hat drei Stellungen. Die mittlere Stellung verriegelt beide Abdeckungen. Wenn Sie die Arretierung in die entgegengesetzte Richtung schieben, können Sie einen der Einbauplätze öffnen.

(1) Vergewissern Sie sich zuerst, daß die interne Batterie in der RAM-Karte eingelegt ist.

(2) Schalten Sie den PC-1360 aus.

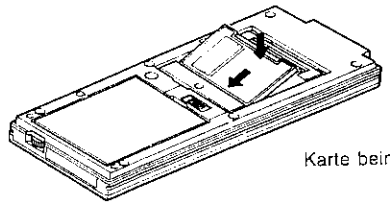
(3) Öffnen Sie einen der Einbauplätze, indem Sie die Arretierung verschieben, und entfernen Sie die Abdeckung. Achten Sie darauf, daß Sie keine Steckerkontakte berühren.



Ansicht der Unterseite des Gerätes

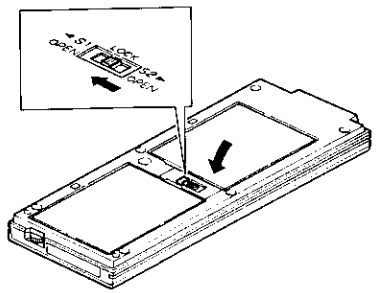
(4) Die Kontaktseite der RAM-Karte in Pfeilrichtung einlegen. Achten Sie darauf, daß Sie die Karte nicht umgekehrt einlegen.

(5) Drücken Sie leicht auf den Batteriedeckels der RAM-Karte, bis die Karte einschnappt.



Karte beim Herunterdrücken

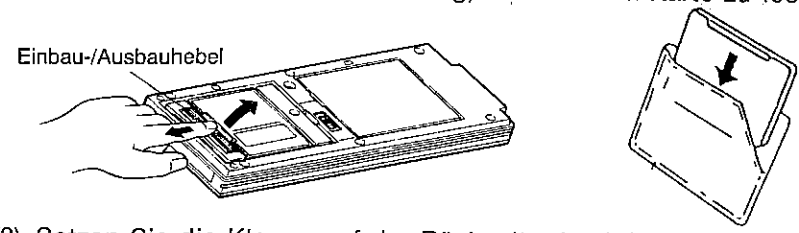
(6) Legen Sie die Abdeckung wieder ein und verriegeln sie. Bitte beachten Sie, daß der PC-1360 nicht funktioniert, wenn die Abdeckung nicht richtig verriegelt ist. Falls Sie es vergessen haben, verriegeln Sie die Abdeckung, schalten das Gerät aus und dann wieder ein.



2. Entnahme der RAM-Karte

(1) Schalten Sie den PC-1360 aus.

(2) Entfernen Sie die Klappe auf der Rückseite des PC-1360 wie oben im Schritt 1 beschrieben. Schieben Sie den Hebel zum Einsetzen/Herausnehmen der Karte in Pfeilrichtung, um die RAM-Karte zu lösen.



(3) Setzen Sie die Klappe auf der Rückseite des PC-1360 wieder ein und schieben Sie die Arretierung wieder in die "LOCK"-Position.

3. Einsatz der RAM-Karte

Die Struktur des Speichereinsatzes hängt von der Einstellung von SET MEM ab. Es gibt drei Möglichkeiten. Siehe dazu das Kommando SET MEM.

Das in der RAM-Karte eingeschriebene Programm bleibt auch dann erhalten, wenn die RAM-Karte aus dem PC-1360 entnommen wird (dank der in der RAM-Karte eingesetzten Batterie). Deshalb kann das gleiche Programm zusammen mit der RAM-Karte erneut in den PC-1360 eingeladen werden.

Der PC-1360 kann zwei RAM-Karten in seinen zwei Einbauplätzen enthalten. Wenn zwei RAM-Karten benutzt werden, achten Sie immer darauf, daß diese wieder in die gleichen Einbauplätze eingelegt werden, in denen sie vorher waren, da die Funktion jeder Karte verschieden ist je nachdem in welchem Einbauplatz sie eingelegt ist. Normalerweise ist das Programm in der Karte im Einbauplatz 1 und die Programmvariablen befinden sich in der Karte im Einbauplatz 2. Was die Optionen für den Gebrauch der Einbauplätze betrifft, sehen Sie nach unter dem Kommando SET MEM. Wenn Sie die Karten verwechseln und in die flaschen Einbauplätze einlegen, erscheint eine Initialisierungs-Meldung.

Die möglichen Einsatzkombinationen der Einbauplätze sind hier unten angegeben:

1/ Einbauplatz 1 = MEM$"C"-Karte
Einbauplatz 2 = irgendeine andere Karte

2/ Einbauplatz 1 = MEM$"B"-Programmseite
Einbauplatz 2 = MEM$"B"-Datenseite

3/ Einbauplatz 1 = MEM$"D"-Programmseite
Einbauplatz 2 = MEM$"D"-Datenseite

Mit einer MEM$"D"-Karte braucht die dazugehörende Karte im anderen Einbauplatz nicht zuerst formatiert zu werden.

Initialisierungs-Meldungen erscheinen, wenn folgende Kombinationen verwendet werden:

1/ Einbauplatz 1 = MEM$"D"-Programmkarte
Einbauplatz 2 = neue Karte

```
MEM$ = "D"
RAM CARD S2 CLEAR O.K. ?
```

2/ Einbauplatz 1 = neue Karte
Einbauplatz 2 = MEM$"D"-Datenseitenkarte

```
MEM$ = "D"
RAM CARD S1 CLEAR O.K. ?
```

Wenn die Karten in dieser Weise zusammen eingesetzt werden und Sie irgendeine Taste sußer der **Y**-Taste drücken, erscheint die Initialisierungs-Meldung:

```
MEM$ = "C"
RAM CARD S1 CLEAR O.K. ?
```

4. Vorsichtmaßnahmen beim Einsatz der RAM-Karte

Die gespeicherten Programme und Daten werden gelöscht, wenn die Batterie in der RAM-Karte ausgewechselt (oder herausgenommen) wird. Wenn also ein wichtiges Programm in der RAM-Karte gespeichert ist, ist es empfehlenswert, das Programm vorher auf Band zu speichern.

Wird die Batterie in der RAM-Karte gewechselt, während die RAM-Karte im PC-1360 eingesetzt ist, so bleibt der Inhalt erhalten.

Drücken Sie nie den ALL RESET-Schalter auf der Rückseite des PC-1360, während eine RAM-Karte eingesetzt ist.

Falls er aus Versehen betätigt wird, werden die in der RAM-Karte gespeicherten Programme und Daten gelöscht. Siehe Kapitel 2 für weitere Einzelheiten über die Initialisierung des PC-1360 und den Schutz des Speicherinhaltes von RAM-Karten.

5. DAS KOPIEREN VON RAM-KARTEN

RAM-Karten, die keine Passwort-geschützten Programme enthalten, können kopiert werden. Die Karte, auf die kopiert werden soll, muß in den Einbauplatz 1 und die zu kopierende Karte in den Einbauplatz 2 eingelegt werden. Beide Karten müssen die gleiche Speichergröße haben.

(1) Schten Sie das gerät aus.

(2) Legen Sie die zu kopierende Karte in den Einbauplatz 1 ein.

(3) Legen Sie Karte, auf die kopiert werden soll, in den Einbauplatz 2 ein.

(4) Schalten Sie das Gerät ein, während Sie auf die **C**-Taste drücken. Die folgende Anzeige erscheint:

```
RAM CARD COPY ?
```

Wenn Sie irgendeine Taste außer der **Y**-Taste drücken, wird das Kopieren unterbrochen und der Rechner geht zur normalen Betriebsanzeige zurück:

```
RUN MODE
>
```

Wenn Sie die **Y**-Taste drücken, wird ein anderes Aufforderungszeichen erzeugt:

```
S1:SOURCE S2:DEST OK  ?
```

Wenn Sie irgendeine andere Taste außer der **Y**-Taste drücken, erscheint die folgende Anzeige und das Gerät sollte ausgeschaltet werden:

```
                    *
```

Drücken Sie die **Y**-Taste, um mit dem Kopieren zu beginnen:

```
COPYING...
```

Nach Beendigung des Kopierens erscheint die folgende Meldung:

```
COPY COMPLETED.
```

6. Gebrauch von RAM-Karten anderer Computer

Die PC-1450 und PC-1460 Programme, die in der Karte enthalten sind, werden automatisch für den PC-1360 umgesetzt. Nach der Umsetzung, wenn der Arbeitsbereich des Systems (etwa 1,5 KB auf der RAM-Karte) nicht erhältlich ist, erscheint die folgende Meldung:

```
MEM$ = "C"
RAM CARD S1 CLEAR O.K.  ?
```

Wenn also eine Zeile 80 Byte überschreitet, wird der ekzessive Teil ignoriert und nach der Umsetzung erscheint eine Meldung:

```
WARNING:
SOME LINE EXCEEDED 79 B
HIT ANY KEY.
```

Wenn die Umsetzung ohne Fehler durchgeführt wird, erscheint die normale Arbeitsanzeige:

```
RUN MODE
>
```

Bitte beachten Sie, daß in dieser Prozedur alle Variablen usw. gelöscht werden.

KAPITEL 9 COMPUTER-ANWEISUNGEN DES PC-1360

Das folgende Kapitel gibt alle Computer-Anweisungen in alphabetischer Reihenfolge an, die auf dem PC-1360 eingesetzt werden. Falls eine vorkommt, wird die kürzeste erkennbare Abkürzung für jeden Anweisungsnamen angegeben. Die Anweisungen sind in drei Hauptblöcke unterteilt:

KOMMANDOS

Kommandos werden gegeben, um Arbeitsbedingungen zu ändern, Dienstprogramme auszuführen oder das Programm zu kontrollieren. Sie werden ohne Zeilennummer eingegeben und durch Betätigen der **ENTER**-Taste ausgeführt.

BEFEHLE

Befehle werden innerhalb eines Programms verwendet. Jeder Befehl bildet eine einzelne BASIC-Anweisung. Der Befehl wird durch eine Zeilennummer links vom Gleichheitszeichen einer BASIC-Anweisung angegeben (außer wenn eine einzelne Programmlinie mehr als einen Befehl enthält).

FUNKTIONEN

Eine Funktion ist ein spezieller Operator, der in einem BASIC-Programm verwendet wird, um eine Variable in eine andere umzuwandeln. Sie erscheint irgendwo auf der rechten Seite des Gleichheitszeichen der BASIC-Anweisung, außer wenn sie direkt mit dem Computer als Taschenrechner eingegeben wird. Einige Funktionen nehmen ein Argument. Die, die es nicht tun, werden in der gleichen Weise behandelt wie einfache Programmvariablen und werden manchmal Pseudovariablen genannt.

Numerische Funktionen bestehen aus einem einzigen Zahlenwert und errechnen ein numerisches Ergebnis. Sie enthalten trigonometrische Funktionen, logarithmische Funktionen und Funktionen, die mit einer ganzen Zahl und Vorzeichenteile einer Zahl arbeiten. Anders als bei einigen Formen des BASIC, wenn der Argumententeil eine Zahl oder eine Einzelvariable ist, braucht das Argument nicht in Klammern zu stehen.

Mit Zeichenfolgen-Funktionen werden Zeichenfolgen des Textes manipuliert. Einige nehmen ein Argument und geben einen numerischen Wert zurück. Einige nehmen ein Zeichenfolgen-Argument und geben eine Zeichenfolge zurück. Einige nehmen einen numerischen Wert und geben eine Zeichenfolge zurück. Einige nehmen ein Zeichenfolgen-Argument und ein oder zwei numerische Argumente und geben eine Zeichenfolge zurück. Anders als bei einigen Formen des BASIC, wenn die Funktion ein einzelnes Argument nimmt und das Argument kein Ausdruck ist, können die Klammern ausgelassen werden.

Einige Instruktionen können sowohl als Kommando als auch als Befehl benutzt werden. Alle Instruktionen müssen in Großbuchstaben eingegeben werden. Es gibt fünf Spezialgruppen von Anweisungen:

BILDSCHIRM-GRAFIK bezogene Anweisungen.

Sie kontrollieren die grafische Ausgabe auf dem Bildschirm des PC-1360.

PLOTTER/DRUCKER-GRAFIK bezogene Anweisungen.

Sie beziehen sich auf die grafischen Charakteristika des CE-140P Farb-Matrixdruckers und des CE-515P/516P Farb-Plotter/ Druckers. Grafische Daten können an den Plotter/Drucker mit Hilfe von grafischen Codes und dem LPRINT-Kommando übertragen werden. Diese Spezialkommandos erleichtern Ihnen die Arbeit und vereinfachen Ihr Programm. Alle Parameterwerte, die für die Grafik bezogenen Kommandos des Plotter/Druckers (außer der mit CIRCLE gegebenen Verhältniszahl) spezifiziert werden, werden bei der Programmausführung auf ganze Zahlen abgeschnitten.

KASSETTENTONBAND bezogene Anweisungen.

Sie kontrollieren das Lesen und Schreiben von Daten auf einem Kassettentonband.

SERIELLER E/A-INTERFACE bezogene Anweisungen.

Diese behandeln die Dateneingabe und -ausgabe über das serielle E/A-Interface.

MASCHINENSPRACHE bezogene Anweisungen.

Diese werden benutzt, um Programme in der Maschinensprache aufzurufen und auszuführen, und um auf einzelne Speicher-Byte zuzugreifen und sie zu lesen.

Die folgenden Abkürzungen werden verwendet, um die Art der Anweisungen in den nachfolgenen Erklärungen zu bewerten.

C=Kommando
V=Verb
F=Funktion
P=Drucker
Gs=Bildschirm-Grafik bezogen
Gp=Plotter/Drucker-Grafik bezogen
T=Kassettenband bezogen
S=Serielles E/A-Interface bezogen
M=Maschinensprache bezogen

ABS C,V,F

FORMAT: ABS(X)

Abkürzung: AB.
Vergleiche:

Wirkung:
Errechnet den absoluten Wert der Zahl X.

Anmerkungen:
Die Zahl X kann jeder beliebige numerische Ausdruck sein.

Beispiel:

```
 5: WAIT 70
10: PRINT "          ZAHL          ABSOLUT"
20: FOR I=-2 TO 2
30: PRINT I, ABS (I)
40: NEXT I
50: END

RUN
                    ZAHL          ABSOLUT
                      -2.               2.
                      -1.               1.
                       0.               0.
                       1.               1.
                       2.               2.

>
```

ACS

C,V,F

FORMAT: 1. ACS(X)

Abkürzung: AC.
Vergleiche: ASN,ATN,COS

Wirkung:
Errechnet den Arkuscosinus von X.

Anmerkungen:
Diese Funktion errechnet den Arkuscosinus des Ausdrucks X in Alt-Grad, Radial-Wert oder Neu-Grad je nach dem Modus, in welchen der Computer mit dem Kommando DEGREE, RADIAN oder GRAD gesetzt wird. Der Wert von X wird auf $-1 \leqq X \leqq 1$ beschränkt.

Beispiel:

```
10: DEGREE
20: PRINT "ARC COS OF 0.5 IS  ";ACS (0.5)
30: PRINT "ARC COS OF 0 IS  ";ACS (0)
40: END

RUN
ARC COS OF 0.5 IS  60.
ARC COS OF 0 IS  90.
>
```

AREAD V

FORMAT: 1. AREAD variable name

Abkürzung: AR.
Vergleiche: Befehl INPUT und die Beschreibung der DEF-Taste in kapitel 6.

Wirkung;
Der AREAD-Befehl wird benutzt, um eine einzige Variable in ein Programm einzulesen, welches mit der **DEF**-Taste gestartet wurde.

Anwendung:
Wenn ein Programm ein Etikett in Form eines Buchstaben zugewiesen wurde, so daß es mit der **DEF**-Taste gestartet werden kann, kann der AREAD-Befehl benutzt werden, um einen einzelnen Startwert, ohne den INPUT-Befehl zu setzen. Der Befehl AREAD muß in der ersten Zeile nach der Zeile mit dem Etikett gesetzt sein. Taucht er an anderer Stelle im Programm auf, so wird er ignoriert. Es können sowohl numerische als auch Zeichenfolgen-Variablen benutzt werden, aber immer nur ein Wert pro Programm.

Um den AREAD-Befehl zu benutzen, geben Sie den gewünschten Wert im RUN-Modus ein und drücken die **DEF**-Taste, gefolgt von dem Buchstaben, der das Programm kennzeichnet. Wird eine Zeichenfolgen-Variable benutzt, so muß diese nicht in Anführungszeichen gesetzt werden.

Wird das Bereitschaftssymbol {>} zu Beginn der Programmausführung angezeigt, so ist die benutzte Variable gelöscht.

Beachten Sie, wie im folgenden Beispiel unter Verwendung von PRINT am Beginn des Programmes Text gespeichert wird.

```
10: "A": PRINT "ABC", "DEFG"
20: "S": AREAD A$: PRINT A$
RUN mode
DEF A → ABC                    DEFG
DEF S → DEFG
```

Wird PRINT Numerischer Ausdruck, Numerischer Ausdruck, Numerischer Ausdruck...., oder PRINT "Zeichenfolge", "Zeichenfolge", "Zeichenfolge" ..... angezeigt, so wird der zuletzt angezeigte Inhalt gespeichert.

Wird PRINT Numerischer Ausdruck; numerischer Ausdruck; Numerischer Ausdruck... angezeigt, so wird der erste (ganz links) angezeigte Inhalt gespeichert.

Wird PRINT "Zeichenfolge"; "Zeichenfolge"; "Zeichenfolge" .... angezeigt, so wird der zuletzt angezeigte Inhalt der "Zeichenfolge" gespeichert.

Beispiele:

```
1Ø: "X":AREAD N
2Ø: PRINT N^2
3Ø: END
```

Die Eingabe von "7 **DEF** X" gibt die Anzeige "49" aus.

ARUN V

FORMAT: 1. ARUN [(ausdruck)]
2. ARUN [(zeichenfolge)]

Abkürzung: ARU.

Vergleiche: AUTOGOTO, RUN

Wirkung:
Stellt den Computer so ein, daß der Programmablauf automatisch beim Einschalten erfolgt.

Anmerkungen:
Wenn ARUN in der ersten Programmanweisung (d.h. in der kleinsten Zeilennummer im Programm) beim Einschalten enthalten ist, läuft das Programm wie wenn ein RUN-Kommando über die Tastatur eingegeben worden wäre.
Dieser Befehl wirkt wie der AUTO GOTO-Befehl, außer daß vor der Programmausführung alle Variablen und Datenfelder gelöacht werden.

Beispiel:

```
 5: ARUN
10: CLS : WAIT 50
20: PRINT "WELCOME TO THE WORLD OF"
30: PRINT "THE COMPUTER"
40: PRINT "YOU HAVE "; MEM ;" BYTES FREE"
50: END
```

Das Programm läuft automatisch bei POWER ON.

ASC

C,V,F

FORMAT: 1. ASC (zeichenfolgen-variable)
2. ASC (zeichenfolge)

Abkürzung: A.
Vergleiche: CHR$

Wirkung:
Berechnet den ASCII-Codewert für das erste Zeichen in der vorgegebenen Zeichenfolge.

Anmerkungen:
Die Zeichenfolge kann als Inhalt einer Zeichenfolgen-Variablen in der Form X$ oder als wirkliche Zeichenfolge in Anführungszeichen, wie z.B. "XXXX", angegeben werden. Nur der Wert des ersten Zeichens in der Zeichenfolge wird berechnet, und zwar unabhängig der Länge. Die Zeichencode-Tabelle finden Sie in Anhang B.

Beispiel:

```
10: INPUT "ENTER A CHARACTER ";A$
20: N= ASC (A$)
30: PRINT "THE ASCII CODE IS ";N
40: GOTO 10
```

[10] Der Benutzer drückt eine Taste, um irgendein Zeichen einzugeben.
[20] ASC findet die Codenummer für diesen Zeichen.
[30] Druckt die Antwort aus.
[40] Wiederholt sich bis der Benutzer das Programm durch Betätigen der BRK -Taste stoppt.

ASN

C,V,F

FORMAT: 1. ASN(X)

Abkürzung:
Vergleiche: ACS, ATN, SIN

Wirkung:
Errechnet den Arkuscosinus von X.

Anmerkungen:
Diese Funktion errechnet den Arkuscosinus des Ausdrucks X in Alt-Grad, Radial-Wert oder Neu-Grad je nach dem Modus, in welchen der Computer mit dem Kommando DEGREE, RADIAN oder GRAD gesetzt wird. Der Wert von X wird auf $-1 \leqq X \leqq 1$ beschränkt.

Beispiel:
```
 5: WAIT 6Ø
1Ø: CLS
2Ø: DEGREE
3Ø: PRINT "ARC SIN", "ANGLE"
4Ø: FOR H = Ø TO1Ø
5Ø: X = H/1Ø
6Ø: DX =   ASN X
7Ø: PRINT  X,DX
8Ø: NEXT H
```

ATN

C,V,F

FORMAT: 1. ATN(X)

Abkürzung: AT.
Vergleiche: ACS, ASN, TAN

Wirkung:
Errechnet den Arkustangens von X.

Anmerkungen:
Diese Funktion errechnet den Arkustangens des Ausdrucks X in Alt-Grad, Radial-Wert oder Neu-Grad je nach dem Modus, in welchen der Computer mit dem Kommando DEGREE, RADIAN oder GRAD gesetzt wird.

Beispiel:

```
1Ø: CLS
2Ø: RADIAN
25: PRINT "TANGENT", "ANGLE"
3Ø: FOR T= Ø TO 2Ø
4Ø: RT=  ATN (T)
5Ø: PRINT T,RT
6Ø: NEXT T
```

AUTOGOTO V

FORMAT: 1. AUTOGOTO (ausdruck)
2. AUTOGOTO (zeichenfolge)

Abkürzung: AU.
Vergleiche: ARUN, GOTO

Wirkung;
Stellt den Computer so ein, daß der Programmablauf automatisch beim Einschalten erfolgt.

Anmerkungen:
Wenn AUTOGOTO in der ersten Programmanweisung (d.h. in der kleinsten Zeilennummer im Programm) beim Einschalten enthalten ist, läuft das Programm wie wenn ein GOTO-Kommando über die Tastatur eingegeben worden wäre. Dieses Kommando ähnelt dem ARUN-Kommando, erlaubt Ihnen aber, das Programm von einer vorgegebenen Zeile an ablaufen zu lassen und löscht nicht alle Variablen und Felder vor der Ausführung. Das AUTOGOTO-Kommando startet nicht den automatischen Programmablauf beim Einschalten.

```
10: AUTOGOTO 60
  :
  :
  :
```

Bei POWER ON beginnt die Ausführung des Programmes ab Zeile 60.

BASIC C

FORMAT: 1. BASIC

Abkürzung: BA.
Vergleiche: TEXT

Wirkung:
Löscht den TEXT-Modus (nur im Programm-Modus erlaubt).

Anwendung:
Die Ausführung dieses Kommandos löscht den Text-Modus und schaltet den Modus in den BASIC-Modus um. Das Bereitschaftssymbol wechselt von "<" nach ">".

Normalerweise wandelt der Wechsel aus dem Text-Modus in den BASIC-Modus den Text im PC-1360 in ein Programm um (interner Code).

Beachten Sie, daß Abkürzungen wie z.B. "P." und "l." nicht in ihre entsprechenden Kommandos übersetzt werden. In diesen Fällen, führen Sie den Cursor in die entsprechende Zeile und drücken die **ENTER**-Taste, um den abgekürzten Code in einen vom PC-1360 erkannten Kommandonamen zu übersetzen. Alle Kommandos oder Formate, die im Text-Modus benutzt werden, aber nicht im PC-1360 vorhanden sind, können nicht ausgeführt werden.

Während einer Programmumformung wird das Zeichen "* *" am Ende der vierten Zeile angezeigt. Wurde ein Passwort gesetzt, erhalten Sie auf das BASIC-Kommando eine Fehlermeldung (ERROR 1).

BEEP C,V

FORMAT: 1. BEEP number
2. BEEP number [, tone [, duration]]

Abkürzung: B.
Vergleiche:

Wirkung:
Erzeugt Pieptöne der vorgegebenen Höhe und Dauer über den internen Lautsprecher des Computers.

Anmerkungen:
Zahl spezifiert die Anzahl der Pieptöne. Der Wert muß positiv sein oder ein Ausdruck bis zu 9.999999999E99.

Die Option tonhöhe spezifiziert die steigende Tonhöhe (Frequenz) des Pieptones im Bereich von 255 bis 0.255 und entspricht einer Frequenz von ungefähr 230 Hz, während 0 einer Frequenz von ungefähr 2.8 kHz entspricht.

Die Option dauer spezifiziert die Dauer des Pieptones im Bereich von 0 bis 65535. Die Einstellung der Piepdauer ändert sich mit dem Parameter tonhöhe. Der gleiche Wert der Dauer erscheint etwas länger bei niedrigen Frequenzen.

Wenn Sie nur zahl vorgeben wollen, benutzen Sie das Kommando in der gleichen Weise wie für den PC-1360, auch wenn der Ton etwas verschieden sein kann, weil die Hardware etwas verschieden ist. Die Ausführung des BEEP-Kommandos kann mit Hilfe der **BRK**-Taste unterbrochen werden.

Die Tabelle hier unten gibt Ihnen Einzelheiten über die eingesetzten Parameter-Kombinationen.

Note	No.	Idealfrequenz	Tatsächliche Frequenz	Toleranz (%)	Tonhöhe	Note	No.	Idealfrequenz	Tatsächliche Frequenz	Toleranz (%)	Tonhöhe
do	3	261.6	261.8	0.08	222	do	27	1046.5	1040.7	−0.56	39
do#	4	277.2	277.7	0.18	208	do#	28	1108.7	1113.0	0.39	35
re	5	293.7	294.3	0.20	195	re	29	1174.7	1174.3	−0.03	32
re#	6	311.1	311.4	0.10	183	re#	30	1244.5	1242.7	−0.14	29
mi	7	329.6	329.0	−0.18	172	mi	31	1318.5	1319.6	0.08	26
fa	8	349.2	348.8	−0.11	161	fa	32	1396.9	1406.6	0.69	23
fa#	9	370.0	371.0	0.27	150	fa#	33	1480.0	1471.2	−0.60	21
so	10	392.0	391.4	−0.20	141	so	34	1568.0	1580.2	0.77	18
so#	11	415.3	414.2	−0.27	132	so#	35	1661.2	1662.3	0.07	16
la	12	440.0	439.9	−0.02	123	la	36	1760.0	1753.4	−0.38	14
la#	13	466.2	465.5	−0.15	115	la#	37	1864.7	1855.1	−0.52	12
ti	14	493.9	494.2	0.06	107	ti	38	1975.5	1969.2	−0.32	10
do	15	523.3	522.4	−0.17	100	do	39	2093.0	2098.4	0.26	8
do#	16	554.4	554.1	−0.05	93						
re	17	587.3	589.9	0.44	86						
re#	18	622.3	624.4	0.34	80						
mi	19	659.3	656.4	−0.44	75						
fa	20	698.5	699.4	0.13	69						
fa#	21	740.0	739.9	−0.01	64						
so	22	784.0	785.3	0.17	59						
so#	23	830.6	825.8	−0.58	55						
la	24	880.0	882.8	0.32	50						
la#	25	932.3	934.3	0.21	46						
ti	26	987.8	992.2	0.44	42						

MIN. 230.6 Hz (255)
MAX. 2844.4 Hz (ø)

$$T=(90+4n)\times 3.9\ (\mu s)$$

$$f = \frac{1}{T} = \frac{1}{(90+4n)\times 3.9}\ (MHz)$$

Beispiel:

```
10: FOR I=1 TO 3
20: FOR J=5 TO 25 STEP 5
30: BEEP I,J,150
40: NEXT J
50: NEXT I
60: END
```

[10] Diese äußere Schleife wird benutzt, um die Anzahl der Pieptöne von 1 auf 3 zu ändern.
[20] Der Zähler der inneren Schleife wird benutzt, um die Tonhöhe zu verändern.
[30] Die BEEP-Anweisung wird 15 mal ausgeführt.

CALL

C,V,M

FORMAT: 1. CALL speicher addresse

Abkürzung: CA.
Vergleiche: PEEK, POKE

Wirkung:
Abrufen und Ausführen von Subroutinen in Maschinensprache.

Anwerkungen:
Ruft eine Subroutine in Maschinensprache ab, beginnend bei der vorgegebenen Speicheradresse, und führt sie aus. Die Adresse ist ein Wert im Bereich von Ø bis 65535 (&Ø-&FFFF in hexadezimal). Normalerweise kann nur auf die Bank Ø zugegriffen werden, ganz gleich welche RAM-Karte gerade benutzt wird.

CHAIN T,V

FORMAT: 1. CHAIN
2. CHAIN ausdruck
3. CHAIN "dateiname"
4. CHAIN "dateiname", ausdruck

Abkürzung: CHA.

Vergleiche: CLOAD, CSAVE, RUN

Wirkung:
Mit dem CHAIN-Befehl wird ein auf Kassette gespeichertes Programm ausgeführt. Er kann nur in Verbindung mit dem Lesen eines Kassettenbandes benutzt werden.

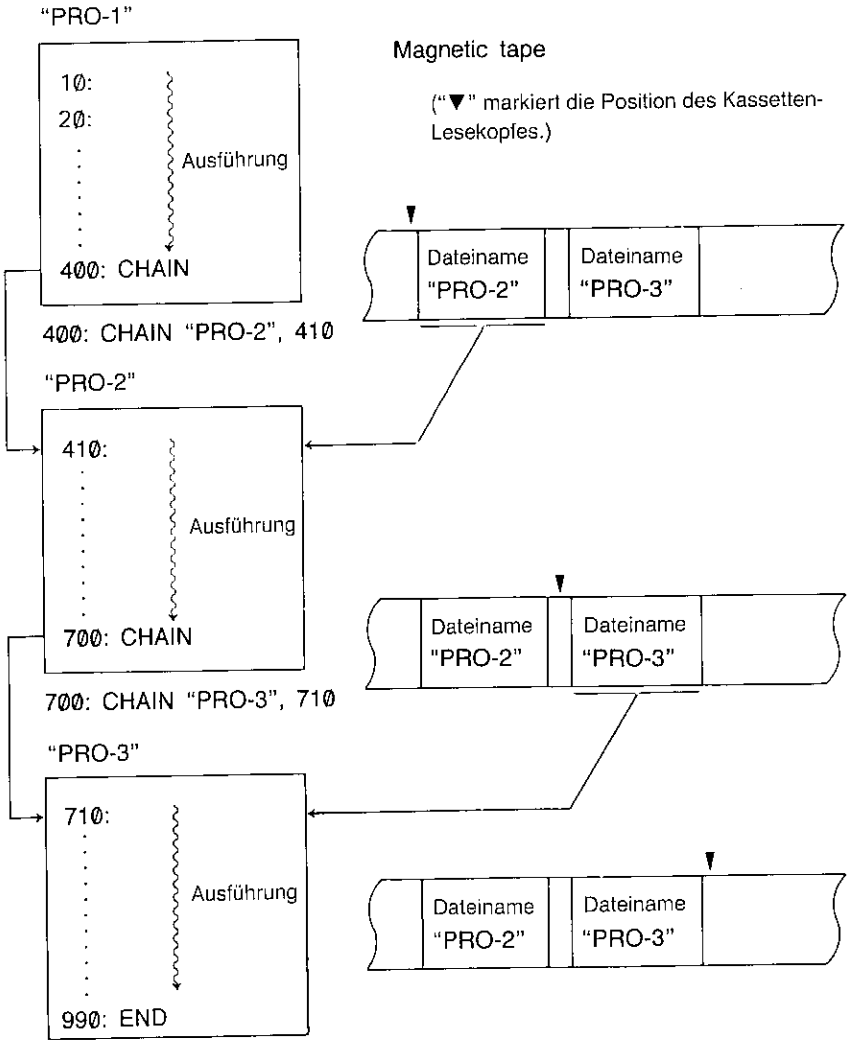
Anmerkungen:
Um den CHAIN-Befehl benutzen zu können, müssen ein oder mehrere Programme auf Kassette abgespeichert sein. Wird dann der CHAIN-Befehl in einem ablaufenden Programm angetroffen, wird das Programm von der Kassette geladen und ausgeführt. Die erste Form von CHAIN lädt das erste auf Kassette gespeicherte Programm und beginnt die Ausführung mit der niedrigsten Zeile dieses Programms. Der Effekt ist der gleiche wie mit dem CLOAD- und dem RUN-Kommando im RUN-Modus.

Die zweite Form von CHAIN lädt das erste auf Kassette gespeicherte Programm und beginnt die Ausführung mit der durch ausdruck angegebenen Zeile.

Die dritte Form von CHAIN sucht die Kassette nach dem mit "dateiname" angebenenen Programm, lädt das Programm und beginnt die Ausführung mit der niedrigsten Zeile.

Die vierte Form von CHAIN sucht die Kassette nach dem mit dateiname angegebenen Programm, lädt das programm und beginnt die Ausführung mit der durch ausdruck angegebenen Zeile.

Nehmen wir z.B. an, daß Sie drei Programmsegmente mit den Namen PRO-1, PRO-2 und PRO-3 haben. Jedes Segment endet mit einem CHAIN-Befehl.



Wenn der Computer während der Ausführung den CHAIN-Befehl antrifft, wird das nächste Segment in den Speicher abgerufen und ausgeführt. In dieser Weise laufen alle Segmente am Ende ab.

Achtung:

Wenn ein Programm, das ein-CHAIN-Kommando enthält, von der Kassette durch ein MERGE-Kommando geladen werden soll, überprüfen Sie zuvor das CHAIN-Kommando auf seine Richtigkeit.

Beispiele:

1Ø: CHAIN

2Ø: CHAIN "PRO-2", 48Ø

[1Ø] Lädt das erste Programm von der Kassette und beginnt die Ausführung mit der niedrigsten Zeile.

[2Ø] Sucht auf der Kassette das Programm PRO-2, lädt es und startet es mit Zeile 480.

CHR$ C,V,F

FORMAT: 1. CHR$ numerischer ausdruck

Abkürzung: CH.
Vergleiche: ASC

Wirkung:

CHR$ ist eine Zeichenfolgen-Funktion, die auf das Zeichen zurückspringt, das dem numerischen Zeichencode seines Arguments entspricht.

Anmerkungen:

Sie finden die Tabelle der Zeichencodes und ihres Verhältnisses zu Zeichen in Anhang B. CHR$ 65 ist gleich "A".

Wenn der Zeichencode 13 angegeben wird, während Sie manuell das CHR$-Kommando ausführen, wird der jeweils folgende Inhalt nicht angezeigt.

Beispiel:
CHR$7Ø + CHR$71 + CHR$13 + CHR$75 + CHR$76
ENTER → FG
Die Zeichen K und L für die Codes 75 und 76 werden nicht an-gezeigt.

CIRCLE

C,V,Gp

FORMAT: 1. CIRCLE (ausdruck 1, ausdruck 2), ausdruck 3
[,ausdruck 4][,ausdruck 5][,ausdruck 6]
[,ausdruck 7][,ausdruck 8][,ausdruck 9]

Abkürzung: CI.

Vergleiche: PAINT, COLOR

Wirkung:
Der CIRCLE-Befehl wird benutzt, um einen Kreis zu zeichnen.

Anwerkungen:
Dieser Befehl ist nur im Grafik-Modus gültig und wird benutzt, um einen Kreis, einen Bogen, einen Kreissektor oder eine Ellipse mit einer Vollinie zu zeichnen.

Mit den Ausdrücken 1 und 2 werden die Koordinaten X bzw. Y in der Mitte eines Kreises angegeben. Die Werte der Ausdrücke 1 und 2 müssen innerhalb des Bereichs von −999 bis 999 liegen.

Mit dem Ausdruck 3 wird der Radius eines Kreises angegeben. Der Wert des Ausdrucks 3 muß innerhalb des Bereichs von 1 bis 499 sein ($1 \leqq$ ausdruck $3 \leqq 499$).

Mit dem Ausdruck 4 wird die Farbe der Linie angegeben. Der Wert des Ausdrucks 4 muß innerhalb des Bereichs von 0 bis 7 im erweiterten Farbcode und von 0 bis 3 in einem anderen erweiterten Farbcode spezifiziert werden. (Siehe den COLOR-Befehl für die von den entsprechenden Werten vorgegebenen Farben.)

Wenn der Ausdruck 4 weggelassen wird, wird der vorherige Wert (d.h. die vorher angegebene Farbe) angenommen.

Mit den Ausdrücken 5 und 6 werden der Anfangswinkel bzw. der Endwinkel eines Bogens oder Kreissektors spezifiziert. Die respektiven Werte der Ausdrücke 5 und 6 müssen innerhalb des Bereichs von −2Ø47 und 2Ø47 liegen. Ein Wert gleich 0 gibt die rechte Seite der Koordinaten an. Ein negativer Wert gibt die Richtung im Gegenuhrzeigersinn an. Ein positiver Wert gibt die Richtung im Uhrzeigersinn an.

Der vorgegebene Wert des Ausdrucks 5 ist 0 Grad und der von Ausdruck 6 ist 360 Grad.

Mit dem Ausdruck 7 wird das folgende Verhältnis angegeben.

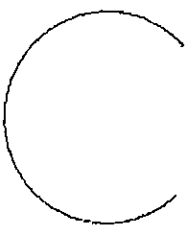
$$\text{Verhältnis} = \frac{\text{ry (Radius in Richtung der Y-Achse)}}{\text{rx (Radius in Richtung der X-Achse)}}$$

Wenn der Wert des Ausdrucks 7 gleich 1 ist, wird ein Kreis gezeichnet. Bei jedem anderen Wert wird eine Ellipse gezeichnet. Der vorgegebene Wert von Ausdruck 7 ist 1.

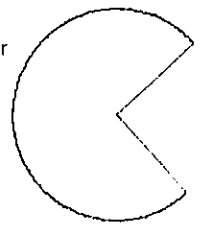
Mit dem Ausdruck 8 wird ein Teilwinkel angegeben. Der Drucker beginnt einen Kreis (Bogen oder Kreissektor) oder eine Ellipse zu zeichnen, wobei dieser in Teileinheiten vom Anfangswinkel bis zum Endwinkel unterteilt wird. Der Wert des Ausdrucks 8 muß innerhalb des Bereichs von 1 bis 2Ø47 liegen. Der vorgegebene Wert von Ausdruck 8 ist 1.

Mit dem Ausdruck 9 wird ein Bogen oder ein Kreissektor angegeben. Ist der Wert von Ausdruck 9 gleich Ø, so wird ein Bogen gezeichnet. Ist der Wert gleich 1, so wird ein Kreissektor gezeichnet. Der vorgegebene Wert von Ausdruck 9 ist Ø.

Ø = Bogen



1 = Kreissektor



Achtung:
Die respektiven Werte der Ausdrücke 5, 6 und 8 sind in Grad ausgedrückt, ganz gleich, welcher Winkel-Modus spezifiziert wird.

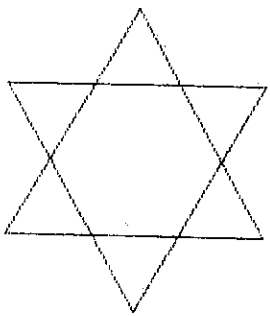
Beispiele:

```
5: OPEN
10: GRAPH
20: CIRCLE (240, −100), 100, 0, 0, 360, 1/2, 10, 0
30: LTEXT
40: LPRINT
50: END
```

[5] Dieser Befehl ist notwendig für den CE-515P/CE-516P.
[20] Verhältniszahl = 0.5
[30] Diese beiden Befehle setzen den Drucker in den Text-Modus zurück und bewegen den Druckkopf an die äußerste linke Position.

```
5: OPEN
10: GRAPH
20: CIRCLE (240, −100), 100, 2, 90, 450, 1, 120, 0
30: CIRCLE (240, −100), 100, 3, −90, 270, 1, 120, 0
40: LTEXT
50: LPRINT
60: END
```

[20] CIRCLE Teilwinkel = 120
[30] Teilwinkel = 120



CLEAR C,V

FORMAT: 1. CLEAR
2. CLEAR variable 1, variable 2, ... variable n

Abkürzung: CL.
Vergleiche: DIM,ERASE

Wirkung:
Mit dem CLEAR-Befehl werden alle in einem Programm benutzten Variablen gelöscht und alle gesetzten Variablen auf Null bzw. leer gesetzt.

Anmerkungen:
Der CLEAR-Befehl setzt Speicherraum frei, der gebraucht wurde, um Variablen zu speichern. Dieser Befehl ist sehr hilfreich, wenn der Speicherbereich eingeschränkt ist und Variablen des ersten Teils eines programms im zweiten Teil nicht mehr gebraucht werden. Der CLEAR-Befehl kann auch am Beginn eines programms benutzt werden, wenn mehrere Programme speicherresident sind und Sie den Bereich frei machen wollen, der für die Ausführung vorheriger Programme benutzt wurde.

CLEAR setzt die Speicherbereiche, die durch die Variablen A—Z, A$—Z$, oder A(1)—A(26) (ohne DIMensionierung) benutzt wurden, nicht frei, wenn diese fest vorgegeben wurden (siehe Kapitel 4).

Das Format 1 löscht alle Variablen und setzt die permanenten Variablen Null oder leer und läßt Zwei-Zeichen-Variablen undefiniert. Das Format 2 erlaubt das Löschen bestimmter Variablen. Im Format 2 können keine Feldnamen angegeben werden. Siehe ERASE-KOmmando für das Löschen bestimmter Felder.

Beispiele:
```
1Ø: A = 5:DIM C(5)
2Ø: CLEAR
```

[2Ø] Setzt den C( ) zugewiesenen Bereich frei und setzt A gleich Null.

CLOAD C,T

FORMAT: 1. CLOAD
2. CLOAD "dateiname"

Abkürzung: CLO.,
Vergleiche: CLOAD?, CSAVE, MERGE, PASS

Wirkung:
Mit dem CLOAD-Kommando wird ein auf Magnetband gespeichertes Programm in den Rechner geladen.

Anwerkungen:
Die erste Form des CLOAD-Kommandos löscht die existierenden Programme aus dem Speicher und lädt das erste Programm auf dem Band, angefangen bei der aktuellen Position.

Die zweite Form des CLOAD-Kommandos löscht den Speicher, sucht das Band nach dem Programm ab, dessen Name mit "dateiname" angegeben ist und lädt das Programm.

Ist der PC-1360 im PROgramm- oder RUN-Modus, so wird das Programm von der Kassette in den Programmspeicher geladen. Ist der PC-1360 im Reserve-Modus, werden die Daten in den Reservespeicher geladen. Sie müssen darauf achten, daß Sie keine Programme in den Reservespeicher und keine Reservedaten in den Programmspeicher laden.

Der Computer kann nicht unterscheiden zwischen einem Programm und einem Reserveprogramm. Wenn ein Programm oder ein Reserveprogramm in den falschen Speicher des Computers geladen wird, kann es sein, daß das System nicht mehr funktionsfähig ist. In diesem Fall, drücken Sie den All Reset-Schalter auf der Rückseite des Computers, um den Zustand zu löschen.

Unterbrechen Sie die Ausführung des Kommandos mit Hilfe der **BRK**-Taste, falls die Suche erfolglos war.

Treten während der Ausführung Fehler auf, wird das geladene Programm nicht zugelassen.

Siehe den Abschnitt über die Bedienung der Kassette im Kapi-tel "Peripheriegeräte".

Beispiele:

CLOAD*

CLOAD "PRO3"**

* Lädt das erste Programm von der Kassette.

**Sucht auf der Kassette das programm 'PRO3' und lädt es ein.

CLOAD? C,T

FORMAT: 1. CLOAD?
2. CLOAD? "dateiname"

Abkürzung: CLO.?
Vergleiche: CLOAD, CSAVE, MERGE, PASS

Wirkung;
Mit dem Kommando CLOAD? wird ein auf Kassette gespeichertes Programm mit einem im Speicher enthaltenen Programm verglichen.

Anmerkungen:
Um zu prüfen, ob ein Programm richtig gespeichert wurde, spulen Sie die Kassette an den Anfang zurück und geben das Kommando CLOAD? ein.

Die erste Form des Kommandos CLOAD? vergleicht das im Speicher enthaltene Programm mit dem ersten Programm auf dem Band, angefangen bei der aktuellen Position.

Die zweite Form des Kommandos CLOAD? sucht nach dem mit "dateiname" angegebenen Programm und vergleicht es dann mit dem Speicherinhalt.

Siehe den Abschnitt über die Bedienung der Kassette im Kapitel "Peripheriegeräte". Beispiele CLOAD? Vergleicht den Speicherinhalt mit dem ersten auf der Kassette im Kapitel "Peripheriegeräte".

Beachten Sie, daß, wenn Sie ein Programm vom Band einlesen lassen, welches von einem PC-1350 Computer gespeichert wurde, der Programm-Code automatisch umgewandelt wird, sodaß er PC-1360 kompatibel ist und CLOAD? nicht zum Vergleichen verwendet werden kann.

Beispiele:
CLOAD?*
CLOAD? "PRO3"**

* Vergleicht den Speicherinhalt mit dem ersten auf der Kassette gefundenen Programm.
** Sucht auf der Kassette anch dem Programm 'PRO3' und vergleicht es dann mit dem Speicherinhalt.

CLOAD M C,T,M

FORMAT: 1. CLOAD M [lade Adresse]
2. CLOAD M "Dateiname" [;lade Adresse]

Abkürzung: CLO.M
Vergleiche: CLOAD, CSAVE M

Wirkung:
CLOAD M wird zum Laden eines Programmes in Maschinensprache verwendet.

Anmerkungen:
CLOAD M lädt ein Programm in Maschinensprache vom Band in den Speicher.

BASIC-Programme, die sich im Speicher befinden, werden nicht überschrieben. Wird die Ladeadresse ausgelassen, wird das in Maschinensprache geschriebene Programm in die Adresse geladen, die zuletzt zum Speichern mit CSAVE M verwendet wurde. Wird die Ladeadresse angegeben, wird das Progamm beginnend mit dieser Adresse geladen. Wird der Dateiname ausgelassen, spult sich das Band zurück und das erste sich auf dem Band befindliche Program wird geladen.

CLOSE C,V,S

FORMAT: 1. CLOSE [#1]

Abkürzung: CLOS.
Vergleiche: OPEN

Wirkung:
Schließt die Leitung zum seriellen E/A-Interface.

Anwerkungen:
Dieses Kommando schließt die Leitung (im Sinn der Software) zum seriellen E/A-Interface, die mit Hilfe des OPEN-Kommandos geöffnet wurde.

Nach diesem Kommando kann also die Ausgabe zum seriellen E/A-Terminal oder die Eingabe vom gleichen Terminal aus nicht mehr durchgeführt werden.

Das # kann ausgelassen werden.

CLS

C,V

FORMAT: 1. CLS

Abkürzung:
Vergleiche: CURSOR

Wirkung:
Der CLS-Befehl löscht das Anzeigefeld.

Anmerkungen:
Der CLS-Befehl löscht das Anzeigefeld und versetzt das Display in die Startposition Ø.

Beispiele:

```
1Ø: WAIT 3
2Ø: INPUT A$
3Ø: FOR B=Ø TO 23
4Ø: CLS
5Ø: CURSOR B, 1
6Ø: PRINT A$
7Ø: NEXT B
8Ø: CLS
9Ø: END
```

Dieses Programm zeigt die Eingabe an und bewegt sie von der linken zur rechten Seite (von der oberen Zeile zur unteren Zeile). In jedem Durchgang der FOR/NEXT-Schleife wird die Anzeige mit dem CLS-Befehl gelöscht, wird die Startposition des Displays mit dem CURSOR-Befehl verschoben und wird der Inhalt von A$ mit Hilfe des PRINT-Befehls angezeigt. Wenn Sie auf diese Weise das Display schreiben und löschen, können Sie auch veranlassen, daß der Bildschirminhalt rollt. (Löschen Sie Zeile 4Ø und führen Sie das Programm aus. Sehen Sie den Unterschied?)

COLOR C,V,Gp

FORMAT: 1. COLOR ausdruck
2. COLOR ausdruck, 7

Abkürzung: COL.
Vergleiche: LLINE, RLINE, CIRCLE, PAINT

Wirkung:
Der COLOR-Befehl wird benutzt, um die Farbe von Zeichen oder Linien anzugeben, die im normalen oder erweiterten Farb-Modus gedruckt werden.

Anmerkungen:
Mit dem Format 1 (COLOR ausdruck) können Sie vier verschiedene Farben angeben, indem Sie einen Wert von 0 bis 3 dem Ausdruck zuweisen.

Mit dem Format 2 (COLOR ausdruck, 7) können Sie acht verschiedene Farben angeben, indem Sie einen Wert von 0 bis 7 dem Ausdruck zuweisen.

Mit dem CE-140P Farb-Plotter/Drucker können beide Formate benutzt werden, um Farben anzugeben. Der CE-515P Farb-Plotter/Drucker jedoch kann nur das Format 1 benutzen. Farbspezifikationen, die für die jeweiligen Drucker anwendbar sind:

CE-140P

Ausdruckwert	Ø	1	2	3	4	5	6	7
Format 1	Schwarz	Purpur	Grün	Rot				
Format 2	Schwarz	Purpur	Rot	Magentarot	Grün	Türkis	Gelb	Weiß

Anmerkung:
Die durch den Wert "7" angegebene Farbe Weiß bezieht sich auf die Farbe des Druckpapiers. Wenn dieser Wert eingegeben wird, bewegen sich der Druckkopf und das Druckpapier, aber es wird nichts ausgedruckt.

CE-515P/CE-516P

Ausdruckswert	Ø	1	2	3
Format 1	Schwarz	Blau	Grün	Rot

Wenn Sie den COLOR-Befehl mit dem Format 2 ausführen, wird der Drucker automatisch in den erweiterten Farb-Modus gesetzt. In diesem Modus können Sie eine gewünschte Farbe unter den oben erwähnten acht Farben mit Hilfe des Befehls LLINE, RLINE, CIRCLE oder PAIN auswählen.

Der Drucker wird aus dem erweiterten Farb-Modus freigegeben, wenn Sie den Hauptschalter des Computers ausschalten.

Beispiel:

```
1Ø: GRAPH
2Ø: COLOR 1, 7
```

[2Ø] Setzt den erweiterten Farb-Modus und gibt gleichzeitig die Farbe "Purpur" an. (In diesem Fall müssen Sie den CE-140P verwenden.)

CONSOLE C,V,S

FORMAT: 1. CONSOLE ausdruck

Abkürzung: CONS.
Vergleiche: OPEN, LPRINT, LLIST

Wirkung:
Setzt bei Datensendung die Anzahl der Spalten pro Zeile.

Anmerkungen:
Dieses Kommando setzt die Anzahl der Spalten pro Zeile für Daten, die vom seriellen E/A-Interface (Terminal) mit Hilfe des LPRINT- oder LLIST-Kommandos gesendet werden.

Der PC-1360 sendet einen Endcode (CR, LF, oder CR + LF), nachdem die vorgegebene Zeile von Daten gesendet worden ist. Gültige Werte für ausdruck sind ganzzahlige Werte im Bereich von 1 bis 160. Ist der Wert von ausdruck größer als 160, wird die Anzahl der Spalten pro Zeile auf 160 begrenzt. Wenn der Wert gleich 0 oder negativ ist, erhalten Sie eine Fehlermeldung (ERROR 3). Die Eingabe eines Wertes zwischen 1 ~ 7 wird automatisch als 7 behandelt.

Wird kein Ausdruck angegeben, wird das Kommando ignoriert und die alte Anzahl von Spalten pro Zeile wird beibehalten. Die Anzahl der Spalten ist 39, wenn die Batterien gewechselt werden oder Sie den All Reset-Schalter drücken.

CONT C

FORMAT: 1. CONT

Abkürzung: C.
Vergleiche: RUN, STOP

Wirkung:
Mit dem CONT-Kommando setzen Sie die Ausführung eines zeit weise unterbrochenen Programms fort.

Anmerkungen:
Wenn die Ausführung eines Programms mit dem Befehl STOP unterbrochen wird, so kann es mit dem Befehl CONT nach dem Aufforderungszeichen fortgesetzt werden.

Wenn ein Programm mit der **BRK**-Taste unterbrochen wird, so kann es mit dem Befehl CONT fortgesetzt werden.

CONT kann auch benutzt werden, um ein Programm fortzusetzen, das mit PRINT oder GPRINT unterbrochen wurde.

Beispiele:
CONT .

Setzt die Ausführung eines unterbrochenen Programms fort.

COS

C,V,F

FORMAT: 1. COS(X)

Abkürzung:
Vergleiche: ACS, SIN, TAN

Wirkung:
Berechnet den Cosinus des Winkels X.

Anmerkungen:
Diese Funktion berechnet den Cosinus des Winkels X, wobei X in Alt-Grad, Radial-Wert oder Neu-Grad ausgedrückt wird, je nachdem, in welchen Modus der Computer mit dem Kommando DEGREE, RADIAN oder GRAD gesetzt wurde.

Beispiel:

```
10: DEGREE
20: PRINT "COS OF 60 IS "; COS (60)
30: PRINT "COS OF 90 IS "; COS (90)
40: END

RUN
COS OF 60 IS 0.5
COS OF 90 IS 0.
>
```

CROTATE

C,V,Gp

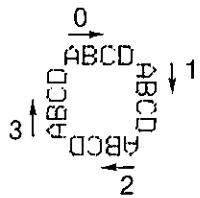
FORMAT: 1. CROTATE ausdruck

Abkürzung: CR.
Vergleiche:

Wirkung:
CROTATE wird benutzt, um die Orientierung und die Druckrichtung von auszudruckenden Zeichen anzugeben.

Anmerkungen:
Dieser Befehl ist nur effektiv, wenn sich der Drucker im Grafik-Modus befindet. Wenn Sie den Wert von ausdruck ändern, können Sie die Druckrichtung und Orientierung von auszudruckenden Zeichen ändern.

Wenn Sie für ausdruck einen Wert von 0 bis 3 eingeben, wird eine der folgenden vier Orientierungen und Druckrichtungen von Zeichen spezifiziert, und der Drucker führt das Drucken in der Pfeilrichtung aus.



Wenn Sie den LTEXT-Befehl ausführen, werden die Orientierung und die Druckrichtung der auszudruckenden Zeichen automatisch auf normal gesetzt, wie wenn der Wert 0 gegeben würde.

Beispiel:

```
 5: OPEN
1Ø: GRAPH
2Ø: GLCURSOR (2ØØ, −3Ø)
3Ø: FOR Z = Ø TO 3
4Ø: CROTATE Z
5Ø: LPRINT "PABCD"
6Ø: NEXT Z
7Ø: LTEXT
8Ø: LPRINT
9Ø: END
```

[5] Dieses Kommando benötigen Sie für den CE-515P/CE-516P.

CSAVE C,T

FORMAT: 1. CSAVE
2. CSAVE "dateiname"
3. CSAVE "passwort"
4. CSAVE "dateiname", "passwort"

Abkürzung: CS.
Vergleiche: CLOAD, CLOAD?, MERGE, PASS

Wirkung:
Mit dem Kommando CSAVE wird ein Programm auf der Kassette abgespeichert.

Anmerkungen:
Die erste Form des Kommandos CSAVE speichert alle Programme im Speicher des Rechners ohne speziellen Dateinamen auf der Kassette ab.

Die zweite Form des Kommandos CSAVE speichert alle Programme im Speicher des Rechners auf der Kassette ab und weist den angegebenen Dateinamen zu.

Die dritte Form des Kommandos CSAVE schreibt alle Programme im Speicher des Rechners ohne speziellen Dateinamen auf der Kassette ab und weist das angegebene Passwort zu. Programme, die mit einem Passwort geschützt sind, können von jedem geladen werden, aber nur von jemandem gelistet oder verändert werden, der das korrekte Passwort kennt. (Siehe PASS-Kommando.)

Die vierte Form des Kommandos CSAVE speichert alle Programme im Speicher des Rechners auf der Kassette ab und weist sowohl den angegebenen Dateinamen wie auch das angegebene Passwort zu.

Wenn der PC-1360 sich im PROgramm- oder RUN-Modus befindet, wird der Inhalt des Programmspeichers auf der Kassette abgespeichert. Wenn der PC-1360 sich im Reserve-Modus befindet, wird der Inhalt in den Reservespeicher abgespeichert. Siehe den Abschnitt über den Gebrauch der Kassette im Kapitel "Peripheriegeräte".

Beispiele:

CSAVE "PRO3", "GEHEIM"

Speichert die jetzt gespeicherten Programme unter dem Namen 'PRO3' mit dem Passwort 'GEHEIM' auf der Kassette ab.

CSAVE M

C,V,T,M

FORMAT: 1. CSAVE M ["Dateiname";] Startadresse, Endadresse

Abkürzung: CS.M
Vergleiche: CLOAD, CLOAD M

Wirkung:
CSAVE M wird zum Speichern eines in Maschinensprache geschriebenen Programmes auf Band verwendet.

Anmerkungen:
CSAVE M speichert ein sich im Speicher befindliches in Maschinensprache geschriebenes Programm in binärem Format auf Band (Cassette).

Wird der Dateiname nicht angegeben, wird das Programm ohne Dateiname auf Band gespeichert. Das Programm wird beginnend von der angegebenen Startadresse bis zur angegebenen Endadresse auf das Band gespeichert.

CSIZE

C,V,Gp

FORMAT: 1. CSIZE ausdruck

Abkürzung: CSI.
Vergleiche:

Wirkung:
CSIZE wird benutzt, um die Größe der auszudruckenden Zeichen anzugeben.

Anmerkungen:
Die Größe der auszudruckenden Zeichen kann angegeben werden, indem Sie einen Wert innerhalb des folgenden Bereichs für ausdruck eingeben.

CE-140P	CE-515P
1 bis 63	1 bis 15

Nehmen wir an, daß der Wert "1" von ausdruck als minimale Zeichengröße gegeben wird, dann sind durch 2,3,4... angegebenen Zeichengrößen... zwei, drei, vier mal so groß wie die minimale Zeichengröße sowie der Zeichenmittenabstand und der Zeilenabstand. Wenn Sie den Netzschalter des Druckers einschalten, ist der Drucker bereit, Zeichen mit CSIZE 2 zu drucken.

Beispiel:
CSIZE 1

Zeichengröße: 0,8 mm(B) × 1,2mm(H) (4 × 6 Schritte)
Zeichenmittenabstand: 1,2 mm (6 Schritte)
Zeilenabstand: 2,4 mm (12 Schritte)

CURSOR C,V

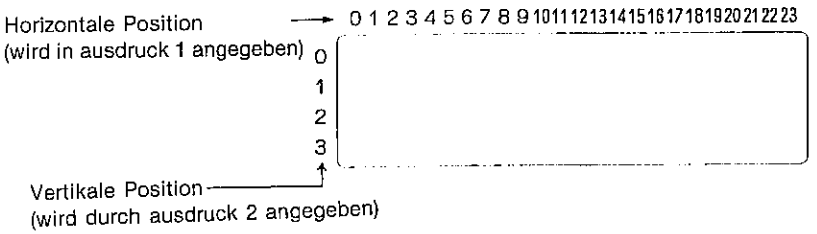
FORMAT: 1. CURSOR ausdruck 1, ausdruck 2
2. CURSOR expression
3. CURSOR

Abkürzung: CU.
Vergleiche: GCURSOR, CLS, INPUT, PRINT, PAUSE

Wirkung:
Dieser Befehl markiert den Startpunkt des Bildschirms in Spalteneinheiten.

Anmerkungen:
Format (1) und (2) geben den Startpunkt des Bildschirmes in Einheiten einer Zeichenposition für den durch das PRINT-Kommando, PAUSE-Kommando usw.

Die Bildschirm-Position wird folgendermaßen mit Format (1) spezifiziert.


Horizontale Position
(wird in ausdruck 1 angegeben)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
0
1
2
3
Vertikale Position
(wird durch ausdruck 2 angegeben)

Jede Position auf dem Bildschirm ist durch einen horizontalen Wert und einen vertikalen Wert spezifiziert. Die Werte von ausdruck 1 und ausdruck 2 geben die horizontle bzw. vertikale Position an.

Der Bereich von ausdruck 1 ist Ø-23 und der Bereich von ausdruck 2 ist Ø-3.

Das Format (3) löscht den Startpunkt des Bildschirmes.

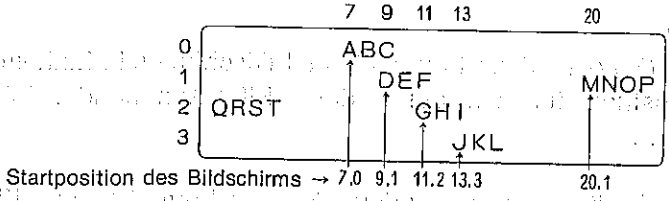
Mit dem CURSOR-Kommando können Sie Text an einem beliebigen Ort auf dem Bilddschirm schreiben ohne den bestehenden Bildschirmtext zu beeinflussen, außer da, wo Zeichen direkt überschrieben werden. Benutzen Sie das CLS-Kommando, um den gesamten Bildschirm zu löschen.

Falls die angezeigten Zeichen die Bildschirmkapazität überschreiten, wird der Bildschirm nach oben gerollt, um alle Zeichen wiedergeben zu können, auch wenn die Startposition mit dem CURSOR-Kommando spezifiziert wurde.

Beispiel:

```
 5: CLS
1Ø: CURSOR 7,Ø:  PRINT  "ABC"
2Ø: CURSOR 9,1:  PRINT  "DEF"
3Ø: CURSOR 11,2: PRINT  "GHI"
4Ø: CURSOR 13,3: PRINT  "JKL"
5Ø: CURSOR 2Ø,1: PRINT  "MNOPQRST"
```

Bei Ausführung erhalten Sie folgendes Bild.



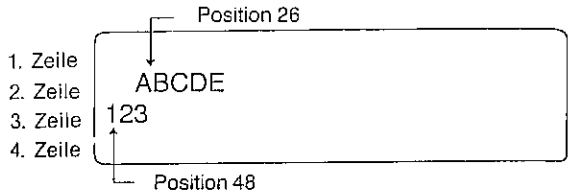
Die Startposition des Bildschirms wird folgendermaßen mit dem Format (2) spezifiziert.

1. Zeile	0 1 2 3 . . . . . 22 23
2. Zeile	24 25 26 . . . . . 46 47
3. Zeile	48 49 50 . . . . . 70 71
4. Zeile	72 73 74 95

Wie oben gezeigt, werden den Positionen auf dem Bildschirm Zahlen von 0 bis 95 zugewiesen, angefangen in der Ecke oben links des Bildschirms bis zur Ecke unten rechts. Der Wert des Ausdruckes in Format (2) spezifiziert die Zahl der Position für die Startposition des Bildschirms. Deshalb müssen Sie sichergehen, daß der Wert des Ausdrucks in Format (2) innerhalb des Bereiches von 0-95 liegt. Wenn dieser Bereich überschritten wird, erhalten Sie eine Fehlermeldung (ERROR 3).

Beispiel:

```
 5: CLS
1Ø: CURSOR 48:  PRINT  123
2Ø: CURSOR 26:  PRINT  "ABCDE"
```



CURSOR kann auch benutzt werden, um den Cursor auf dem Bildschirm zu positionieren und ein INPUT-Kommando einzugeben. Nachdem das INPUT-Kommando ausgeführt ist, bleibt aber der Cursor unverändert an der durch das CURSOR-Kommando angegebenen Stelle erhalten, so daß ein PRINT-Kommando den EINGABE-Text überschreibt.

Beispiel:

```
10: CLS
20: CURSOR  0,2
30: INPUT  "DATA=";A
40: PRINT A/2
```

[30,40] Die in Zeile 20 definierte Position ist sowohl für Zeile 30 als auch für Zeile 40 gültig.

```
10: CLS
20: CURSOR  0,2
30: INPUT  "DATA=";A
35: CURSOR
40: PRINT A/2
```

[35] Die in Zeile 20 definierte Position wird gelöscht.

DATA V

FORMAT: 1. DATA ausdruckliste
Wo: Ausdruckliste: ausdruck
oder: ausdruck, ausdruckliste ist

Abkürzung: DA.
Vergleiche: READ, RESTORE

Wirkung:
DATA wird benutzt, um Daten für READ bereit zu stellen.

Anmerkungen:
Soll ein Feld mit Anfangswerten gefüllt werden, ist es empfehlenswert, die Werte in einem DATA-Befehl aufzulisten und sie dann mit einer FOR...NEXT-Schleife und in einer READ-Anweisung in das Feld einzulesen. Wird der erste READ-Befehl ausgeführt, liest die erste READ-Anweisung den ersten Wert. Mit jedem darauf folgenden READ-Befehl werden die nachfolgenden Werte in der Reihenfolge gelesen, in der sie im Programm erscheinen, ohne Rücksicht zu nehmen auf die Anzahl der in jedem DATA-Befehl definierten Werte oder auf der Anzahl der benutzten DATA-Anweisungen.

DATA-Anweisungen haben keinen Einfluß, wenn sie im Laufe der normalen Ausführung des Programms angetroffen werden, so daß sie an jeder Stelle des Programms eingefügt werden können. Viele Programmierer setzen sie gerne hinter der Zeile, in der der dazugehörende READ- Befehl steht. Wenn nötig, können die Werte in einer DATA-Anweisung mit der RESTORE-Anweisung ein zweites Mal gelesen werden.

Beispiele:
```
1Ø: DIM B(1Ø)
2Ø: WAIT 128
3Ø: FOR I = 1 TO Ø
4Ø: READ B(I)
5Ø: PRINT B(I)
6Ø: NEXT I
```

```
7Ø: DATA 1Ø, 2Ø, 3Ø, 4Ø, 5Ø, 6Ø
8Ø: DATA 7Ø, 8Ø, 9Ø, 1ØØ
9Ø: END
```

[1Ø] Initialisierung eines Feldes
[4Ø] Lädt die Werte aus der DATA-Anweisung in das Feld B( ). B(1) wird gleich 1Ø, B(2) wird gleich 20, B(3) wird gleich 30 usw. gesetzt.

DEG

C,V,F

FORMAT: 1. DEG dd.mmssrr

Abkürzung:
Vergleiche: DMS

Wirkung:
Schaltet Winkelwerte, die in Grad, Minuten und Sekunden angegeben sind, in Dezimal-Grad um.

Anmerkungen:
Der Winkel muß im Format dd.mmssrr angegeben werden, wobei dd Grad (unbegrenzter Bereich) ist, mm Minuten im Bereich von ØØ bis 59 ist, ss Sekunden im Bereich von ØØ bis 59 ist, und rr der Rest des Bereichs von ØØ bis 99 ist. Bitte beachten Sie, daß der Grad-Teil vom Rest durch einen Dezimalpunkt getrennt ist. Vor dem Dezimalpunkt muß immer eine Zahl angegeben werden, wie z.B. .5 als Ø,5.

Beispiel:
```
1Ø: X = DEG 5Ø.3ØØØØØ
2Ø: PRINT X
3Ø: END

RUN
                5Ø.5
```

DEGREE

C,V

FORMAT: 1. DEGREE

Abkürzung: DE.
Vergleiche: GRAD, RADIAN

Wirkung:
DEGREE wird benutzt, um Winkelwerte auf Alt-Grad umzuschalten.

Anmerkungen:
Der PC-1360 kann drei Formen der Winkeldarstellung verarbeiten: Alt-Grad (dezimal), Radial-Werte und Neu-Grad. Diese Formen werden für die Darstellung der Werte der Argumente von den SINus-, COSinus- und TANgens-Funktionen und der Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens benutzt.

Die DEGREE-Funktion schaltet die Form für alle Winkelwerte auf Alt-Grad um, bis Sie GRAD oder RADIAN einsetzen. Die DMS- und DEG-Funktionen werden benutzt, um Winkelwerte von Alt-Grad in Grade, Minuten, Sekunden und umgekehrt umzuformen.

Beispiele:
```
1Ø: DEGREE
2Ø: X = ASN 1
3Ø: PRINT X
```

[2Ø] X hat jetzt den Wert 9Ø, d.h. 9Ø Grad, dem Arcussinus von 1,.

DELETE C

FORMAT: 1. DELETE zeile#
2. DELETE zeile#,
3. DELETE zeile#, zeile#
4. DELETE , zeile#

Abkürzung: DEL.
Vergleiche: NEW

Wirkung:
Löscht die angegebenen Programmzeilen im Speicher.

Anmerkungen:
DELETE zeile# löscht nur die angegebene Programmzeile. DELETE zeile#, löscht die Programmzeilen von der angegebenen Zeilennummer bis zur höchsten Programmzeile im Speicher. DELETE zeile#, zeile# löscht alle Programmzeilen zwischen der ersten angegebenen Zeilennummer (niedrigster Wert) und der zweiten angegebenen Zeilennummer (höherer Wert). DELETE, zeile# löscht die Programmzeilen von der niedrigsten Zeilennummer im Speicher bis zur angegebenen Zeilennummer.

Wenn Sie DELETE im RUN-Modus benutzen, erhalten Sie die Fehlkermeldung ERROR 9. Falls Sie ein Passwort eingesetzt haben, passiert nichts und das Aufforderungszeichen erscheint. Wenn der Speicher mehr als ein Programm enthält, weil das MERGE-Kommando eingesetzt wird, arbeitet DELETE auf das zuletzt gemischte Programm. Als Zeilennummern sind nur die Ziffern Ø bis 9 gültig. Wenn Sie eine Zeile angeben, die nicht existiert, erhalten Sie eine Fehlermeldung. Wenn Sie eine Startzeilennummer angeben, die größer ist als die Endzeilennummer, erhalten Sie auch eine Fehlermeldung.

Um das ganze Programm zu löschen, benutzen Sie das NEW-Kommando.

Beispiel:

DELETE 15Ø*
DELETE 2ØØ,**
DELETE 5Ø, 15Ø***
DELETE ,35****

* Löscht nur die Zeile 15Ø.
** Löscht alles von Zeile 2ØØ bis zur höchsten Zeilennummer.
*** Löscht alle Zeilen zwischen der Zeile 5Ø und der Zeile 15Ø.
**** Löscht alles von der niedrigsten Zeilennummer bis zur Zeile 35.

DIM

C,V

FORMAT: 1. DIM dim liste

Wo: dim liste	dimensionierung oder dimensionierung, dim liste
und: dimensionierung	numerische dim oder zeichenfolgen-dim
und: numerische dim	numerischer name (größe)
und: zeichenfolgen-dim	zeichenfolgen-name (größe) oder zeichenfolgen-name (größe)*länge
und: numerischer name	erlaubter numerischer variablenname
und: zeichenfolgen-name	erlaubter name der zeichenfolgenvariablen
und: größe	größe, oder größe, größe
und: größe	zahl der elemente
und: länge	länge einer zeichenfolge in einem zeichenfolgen-feld

Abkürzung: D.
Vergleiche: ERASE,CLEAR

Wirkung:
DIM wird benutzt, um Speicher für numerische und Zeichenfolgen-Feldvariablen frei zu halten.

Anmerkungen:
Außer für Felder der Form A( ), A$( ), Zwei-Zeichen ( ) und Zwei-Zeichen$( ) muß jedes andere Feld mit einem DIM-Befehl initialisiert werden, um den nötigen Speicherraum bereit zu halten.

Ein Feld kann maximal zwei Dimensionen haben; Die maximale Größe einer Dimension ist 255. Zu der angegebenen Zahl von Elementen, die im DIM-Befehl definiert ist, addiert sich ein "nulltes" Element. Z.B. hat das Feld DIM B(3) die Elemente B(Ø), B(1), B(2) und B(3). In zweidimensionalen Feldern hat es eine extra "nullte" Reihe und Spalte.

In Zeichenfolgen-Feldern kann zusätzlich zur Anzahl der Elemente die Größe der einzelnen Zeichenfolgen-Elemente vorgegeben werden. Zum Beispiel reserviert DIM B$(3)*12 Speicherraum für 4 Zeichenfolgen mit je 12 Zeichen Länge. Wird die Länge der Zeichenfolgen nicht vorgegeben, kann jede Zeichenfolge maximal 16 Zeichen enthalten.

Bei der Initialisierung eines Feldes werden alle Werte eines numerischen Feldes gleich Null und eines Zeichenfolgen-Feldes gleich leer gesetzt.

Was die DIMensionierung der Felder A und A$ betrifft, informieren Sie sich im Abschnitt, in dem die Variablen erklärt werden.

Feld-Variablen können mit dem CLEAR- oder ERASE-Kommando gelöscht (oder undefiniert gesetzt) werden.

Wenn das Programm mit dem RUN-Kommando gestartet wird, werden die Feld-Variablen automatisch gelöscht.

Eine Variable kann nur einmal angegeben werden und jeder Versuch einer erneuten Angabe innerhalb des Programms, ohne zuerst das CLEAR- oder ERASE-Kommando durchzuführen, hat Fehlermeldungen zur Folge. Seien Sie also vorsichtig, wenn Sie ein Programm mit Hilfe des GOTO-Kommandos oder der DEF-Taste ausführen, daß Sie nicht die gleiche DIM-Anweisung noch einmal ausführen, außer wenn Sie zuerst CLEAR ausgeführt haben.

Beispiele:

```
1Ø: DIM B(1Ø)
2Ø: DIM C$(4, 4)*1Ø
```

[1Ø] Reserviert Speicherraum für ein numerisches Feld mit 11 Elementen.

[2Ø] Reserviert Speicherraum für ein zweidimensionales Zeichenfolgen-Feld mit 5 Reihen und 5 Spalten; jede Zeichenfolge enthält maximal 1Ø Zeichen.

DMS C,V,F

FORMAT: 1. DMS winkel

Abkürzung: DM.
Vergleiche: DEG

Wirkung:
Schaltet einen Winkel, der in Alt-Grad angegeben ist, in Grad, Minuten und Sekunden.

Anmerkungen:
Der Winkel springt wieder in das Format dd.mmssrr zurück, wobei dd Grad (unbegrenzter Bereich) ist, mm Minuten im Bereich von 00 bis 59 ist, ss Sekunden im Bereich von ØØ bis 59 ist, und rr der Rest des Bereichs von 00 bis 99 ist. Beachten Sie, daß der Grad-Teil vom Rest durch einen Dezimalpunkt getrennt ist. Der Winkel muß in Graden als Dezimalzahl dd. angegeben werden.

Beispiel:
```
1Ø: X = DMS 5Ø.5
2Ø: PRINT X
3Ø: END

RUN
                5Ø.3
```

END

V

FORMAT: 1. END

Abkürzung: E.
Vergleiche:

Wirkung:
Der END-Befehl signalisiert das Ende eines Programms.

Anmerkungen:
Sind mehrere Programme in den Speicher geladen, so ist es nötig, den einzelnen Programmen Endmarken zu geben, damit bei der Ausführung eines Programms der Rechner nicht in ein anderes hineinläuft. Dies wird mit dem END-Befehl als letzte Anweisung im Programm erreicht.

Beispiele:
```
1Ø: PRINT "HELLO"
2Ø: END
3Ø: PRINT "GOODBYE"
4Ø: END
```

Mit diesen Programmen im Speicher druckt ein Kommando 'RUN 1Ø' das Wort 'HELLO', aber nicht das Wort 'GOODBYE'.Dieses wird auf das Kommando 'RUN 3Ø' ausgegeben.

ERASE C,V

FORMAT: 1. ERASE feld 1, feld 2,...feld n

Abkürzung: ER.
Vergleiche: CLEAR

Wirkung:
Löscht die angegebenen Felder.

Anmerkungen:
ERASE löscht die angebenenen Felder. Der speicherresidente Feldbereich A(26) kann nicht gelöscht werden. Einzelne Feldelemente können nicht gelöscht werden, es wird nur das ganze Feld gelöscht und sein Speicherbereich wird frei. Wenn Sie alle Variablen im Programm löschen wollen, benutzen Sie CLEAR. Um eine Feldgröße neu zu definieren, müssen Sie sie zuerst mit ERASE löschen und dann in einer DIM-Anweisung neu definieren.

Beispiel:
```
1Ø: DIM AA(1Ø)

2ØØ: ERASE AA
```

EXP

C,V,F

FORMAT: 1. EXP(X)

Abkürzung: EX.
Vergleiche: LN

Wirkung:
Errechnet den Wert der Exponentialfunktion e hoch X.

Anmerkungen:
Der Ausdruck X muß zwischen +230.2585092 und −227.9559242. liegen. Für Werte, die darunter liegen, wird 0 angegeben. Um eine andere Zahlenbasis in die Potenz zu erheben, benutzen Sie die Funktion "^". Der Computer behält den Wert von e als 2.718281828.

Beispiel:
EXP(1Ø)
22Ø26.46579

Druckt den Wert von e hoch 1Ø.

FOR V

FORMAT: 1. FOR numerische variable = ausdruck 1 TO ausdruck 2
2. FOR numerische variable = ausdruck 1 TO ausdruck 2 STEP ausdruck 3

Abkürzung: F. STE.

Vergleiche: NEXT

Wirkung:
FOR wird in Verbindung mit NEXT benutzt, um eine bestimmte Operation mehrmals zu wiederholen.

Anmerkungen:
FOR und NEXT schließen eine Gruppe von Anweisungen ein, die wiederholt werden sollen. Wird diese Gruppe das erste Mal ausgeführt, so hat die Schleifenvariable (die Variable, die unmittelbar nach FOR genannt wird) den Wert von ausdruck 1. Erreicht die Ausführung nun den NEXT-Befehl, so wird die Schleifenvariable um die Schrittgröße erhöht und dann mit dem ausdruck 2 verglichen. Ist der Wert der Schleifenvariablen kleiner oder gleich dem ausdruck 2, so wird die eingeschlossene Gruppe ein weiteres Mal ausgeführt, beginnend mit dem Befehl hinter FOR. In der ersten Form des Befehls ist die Erhöhung (STEP) gleich 1. In der zweiten Form wird die Erhöhung durch den ausdruck 3 vorgegeben. Ist der Wert der Schleifenvariablen größer als der ausdruck 2, so wird die Ausführung mit der Anweisung unmittelbar hinter NEXT fortgesetzt. Da der Vergleich am Ende durchgeführt wird, werden die Anweisungen innerhalb eines FOR/NEXT-Paares mindestens einmal ausgeführt.

Der Wert von ausdruck 1, ausdruck 2 und ausdruck 3 muß zwischen −9.999999999E99 und 9.999999999E99 liegen. Wird der ausdruck 3 gleich Null gesetzt, wird die FOR/NEXT in einer unendlichen Schleife ausgeführt. Die Schleifenvariable kann in der umschlossenen Gruppe von Anweisungen benutzt werden, z.B. als ein Feld-Index, man sollte aber vorsichtig sein, wenn der Wert der Schleifenvariablen verändert wird.

Programme dürfen niemals so geschrieben werden, daß von einem Befehl außerhalb eines FOR/NEXT-Paares in eine solche hineingesprungen wird. Programme sollten auch nicht aus einem FOR/NEXT-Paar herausspringen. Und zwar weil die Ausführung der Schleife nicht beendet wird, auch nicht beim Herausspringen. Sie sollten eine FOR/NEXT-Schleife immer über die NEXT-Anweisung verlassen. Der Schleifenvariablenwert muß dazu nur größer als ausdruck 2 gesetzt werden.

Die Gruppe von Anweisungen, die von dem FOR/NEXT-Paar eingefasst werden, kann ein weiteres Paar von FOR/NEXT-Anweisungen enthalten, das eine andere Schleifenvariable haben muß, solange das eingeschlossene Paar vollständig eingeschlossen ist. Wenn also in der FOR-Anweisung in der Gruppe auftaucht, so muß auch der dazugehörende NEXT-Befehl eingeschlossen sein. So können bis zu fünf FOR/NEXT-Paare ineinander "geschachtelt" sein. Ein illegales Herausspringen aus einer inneren Schleife hat die Fehlermeldung ERROR 5 (Schachtelungsfehler) zur Folge.

Beispiele:

```
1Ø: FOR I = 1 TO 5
2Ø: PRINT I
3Ø: NEXT I
4Ø: FOR N = 1Ø TO Ø STEP -1
5Ø: PRINT N
6Ø: NEXT N
7Ø: FOR N = 1 TO 1Ø
8Ø: X = 1
9Ø: FOR F = 1 TO N
1ØØ: X = X*F
11Ø: NEXT F
12Ø: PRINT X
13Ø: NEXT N
```

[1Ø—3Ø] Diese Gruppe von Anweisungen druckt die Zahlen 1, 2, 3, 4, 5 aus.
[4Ø—6Ø] Diese Gruppe von Anweisungen zählt rückwärts von 10, 9 ..... 0.
[7Ø—13Ø] Diese Gruppe von Anweisungen berechnet und druckt die Fakultäten N der Zahlen von 1 bis 1Ø aus.

GCURSOR C,V,Gs

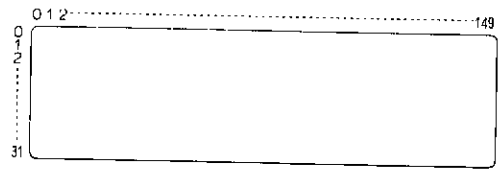
FORMAT: 1. GCURSOR (ausdruck 1, ausdruck 2)

Abkürzung: GC.
Vergleiche: GCURSOR, GPRINT

Wirkung:
Definiert die Anfangasposition der Anzeige in Dot-(Punkt) Einheiten.

Anmerkungen:
Definiert den Startpunkt einer Bildschirmausgabe als Punkte. Anwendung Das GCURSOR-Kommando gibt den Startpunkt des Bildschirmes für das GPRINT-Kommando. (Der Grafik-Cursor wird an die an-gegebene Position gesetzt.)

Der Bildschirm ist aus 15Ø horizontalen und 32 vertikalen Punkten aufgebaut. Jedem dieser Punkte wird eine Nummer von Ø bis 149 in der X-Richtung und von Ø bis 31 in der Y-Richtung zugewiesen. Der Startpunkt des Bildschirmes wird angegeben, wenn die Nummer in der X-Richtung mit ausdruck 1 im obigen Format und die Nummer in der Y-Richtung mit ausdruck 2 angeben.



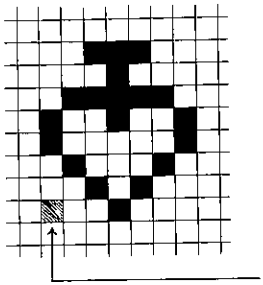
Die Werte von ausdruck 1 und ausdruck 2 können zwischen −32768 und +32767 liegen. Wenn jedoch diese Werte für ausdruck 1 und ausdruck 2 die Bereiche von Ø-149 bzw. Ø-31 überschreiten, befilndet sich die angegebene Position außerhalb der Grenzen des Bildschirmes und eine Position, die in Wirklichkeit nicht existiert, wird für den Startpunkt des Bildschirmes angegeben.

Der Startpunkt des Bildschirmes wird auf (Ø,7) gesetzt, wenn Sie das RUN- oder CLS- Kommando ausführen oder wenn Sie SHIFT und CA drücken. Wind ein Programm mit dem GOTO-Kommando oder einer definierten Taste gestartet, bleibt der Wert für die Y-Richtung erhalten. Die X-Richtung wird auf Ø zurückgesetzt.

Beispiel:

1Ø: GCURSOR (71,2Ø)
2Ø: GPRINT "1824458F452418"

Bei der Ausführung dieses Programms wird folgendes Bild in der Mitte des Bildschirmes wiedergegeben. (Der dunkle Teil wird nicht angezeigt.)



Im obigen Programm ist dieser dunkle Teil der Startpunkt des Bildschirmes (71,2Ø).

GLCURSOR C,V,Gp

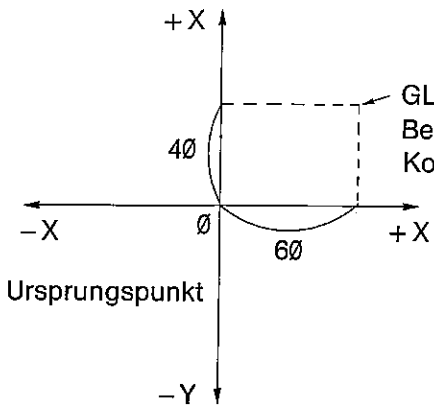
FORMAT: 1. GLCURSOR (ausdruck 1, ausdruck 2)

Abkürzung: GL.
Vergleiche:

Wirkung:
GLCURSOR wird benutzt, um den Zeichenstift zu bewegen.

Anmerkungen:
Dieser Befehl ist nur im Grafik-Modus gültig und wird benutzt, um den Zeichenstift in der X-Achse oder Y-Achse vom Ursprungspunkt der Koordinaten aus zu bewegen.

Der Zeichenstift bewegt sich bis zum Koordinatenpunkt, der mit ausdruck 1 (Koordinate X) und ausdruck 2 (Koordinate Y) angegeben wird. Der Wert für jeden ausdruck muß zwischen −999 und 999 liegen. Die minimale Bewegung in beiden Richtungen beträgt 0,2 mm.



GLCURSOR (6Ø, 4Ø)
Bewegt den Zeichenstift bis zu den Koordinatenpunkten (X = 6Ø, Y = 4Ø).

Achtung:
Das Bewegen des Zeichenstiftes in der +Y-Richtung bedeutet, daß der Papiervorschub in umgekehrter Richtung erfolgen muß.
Siehe die Bedienungsanleitung des Druckers für den "Schneide"-Bereich.

Beispiel:
1Ø: GRAPH
2Ø: GLCURSOR (6Ø, 4Ø)

[2Ø] Bewegt den Zeichenstift zu den Koordinatenpunkten (X = 6Ø, Y = 4Ø).

GOSUB V

FORMAT: 1. GOSUB ausdruck

Abkürzung: GOS.
Vergleiche: GOTO, ON...GOSUB, ON...GOTO, RETURN

Wirkung:
GOSUB wird benutzt, um eine BASIC-Subroutine auszuführen.

Anmerkungen:
Wenn die gleiche Gruppe von Anweisungen in einem Programm mehrere Male ausgeführt werden soll oder wenn diese Gruppe von Anweisungen in verschiedenen Programmen eingesetzt werden soll, so empfiehlt sich die Verwendung der BASIC-Subroutinen mit Hilfe von GOSUB und RETURN.

Die Gruppe von Anweisungen wird im Programm an der gleichen Stelle eingeschlossen, an der sie normalerweise nicht erreicht werden kann. Eine häufige Stelle folgt der END-Anweisung, die das Ende des Hauptprogramms angibt. An diesen Stellen des Hauptprogramms, an denen Subroutinen ausgeführt werden sollen, fügen Sie die GOSUB-Anweisung mit der Zeilennummer der Startzeile der Subroutine als ausdruck ein. Die letzte Zeile der Subroutine muß einen RETURN-Befehl enthalten. Wird der GOSUB-Befehl nun ausgeführt, so springt der PC-1360 an die angegebene Zeilennummer und verarbeitet die Anweisungen, bis ein RETURN-Befehl erreicht wird. Der Rechner springt dann in die Anweisung hinter dem GOSUB-Befehl zurück.

In einer Subroutine kann ein GOSUB vorkommen. Maximal können 1Ø Subroutinen ineinander "geschachtelt" werden.

Der ausdruck in einer GOSUB-Anweisung darf kein Komma enthalten, z.B. 'A(1, 2)' kann nicht benutzt werden. Da eine ON nen an bestimmten Stellen im Programm auszuwählen, besteht normalerweise der ausdruck nur aus der gewünschten Zeilennummer. Wird ein numerischer Wert benutzt, so muß er als ein gültige Zeilennummer bewertet werden, d.h. zwischen 1 und 65279, oder Sie bekommen die Fehlermeldung ERROR 4.

Beispiel:

```
 1Ø: GOSUB 1ØØ
 2Ø: END
1ØØ: PRINT "HELLO"
11Ø: RETURN
```

Bei der Ausführung des Programms druckt dieser Befehl das Wort "HELLO" einmal aus.

GOTO

BEI VERWENDUNG ALS KOMMANDO: **C**

GOTO kann sowohl als Kommando als auch als Befehl eingesetzt werden.

FORMAT: 1. GOTO ausdruck

Abkürzung: G.
Vergleiche: RUN

Wirkung:
Mit dem Kommando GOTO wird die Ausführung eines Programms begonnen.

Anmerkungen:
Das Kommando GOTO kann anstelle des Kommandos RUN benutzt werden, um die Ausführung eines Programms an der durch ausdruck angegebenen Zeilennummer zu starten.

GOTO unterscheidet sich in acht Punkten vom RUN-Kommando:

1) Der Wert des WAIT-Intervalles wird nicht zurückgesetzt.

2) Wurde die Anzeige durch USING-Anweisungen formatiert, wird sie nicht gelöscht.

3) Variablen und Felder bleiben erhalten.

4) Der PRINT = LPRINT-Status wird nicht zurückgesetzt.

5) Der READ-Zeiger wird nicht zurückgesetzt.

6) Die Cursor-Spezifikationen bleiben erhalten.

7) Die horizontale Bewegung des grafischen Cursors wird mit Ø angegeben. Die Einstellung für eine vertikale Bewegung wird beibehalten.

8) Der serielle E/A-Anschluß ist nicht geschlossen.

Die Ausführung eines Programms mit dem Kommando GOTO ist die gleiche wie die Ausführung mit der **DEF**-Taste.

Beispiele:
GOTO 1ØØ

Die Ausführung des Programms wird bei Zeile 1ØØ gestartet.

GOTO WENN SIE GOTO ALS BEFEHL BENUTZEN V

WENN SIE GOTO ALS BEFEHL BENUTZEN:

FORMAT: 1. GOTO ausdruck

Abkürzung: G.
Vergleiche: GOSUB, ON...GOSUB, ON...GOTO

Wirkung:
GOTO wird zum Verlagern des Programmablaufes in die angegebene Zeile verwendet.

Anmerkungen:
GOTO veranlaßt den Rechner, von einer bestimmten Stelle in einem BASIC-Programm an eine andere Stelle zu springen. Anders als bei GOSUB "erinnert" sich GOTO nicht, von wo dieser Sprung ausgeführt wurde.

Der ausdruck in einer GOTO-Anweisung darf kein Komma, wie z.B. in 'A(1, 2)' enthalten. Da eine ON.... GOTO-Struktur vorhanden ist, um verschiedene Punkte an bestimmten Stellen im Programm auszuwählen, besteht normalerweise der ausdruck nur aus der gewünschten Zeilennummer. Wird ein numerischer Wert benutzt, so muß er als eine gültige Zeilennummer bewertet werden, d.h. zwischen 1 und 65279, oder Sie bekommen die Fehlermeldung ERROR 4.

Gute Programme laufen normalerweise von Anfang bis Ende ohne Sprünge, abgesehen von Subroutinen, die während des Programms ausgeführt werden. Der prinzipielle Gebrauch des GOTO-Befehls ist deshalb in der Anweisung IF....THEN zu finden.

Beispiele:

```
1Ø: INPUT A$
2Ø: IF A$="Y" THEN GOTO 5Ø
3Ø: PRINT "NO"
4Ø: GOTO 6Ø
5Ø: PRINT "YES"
6Ø: END
```

Dieses Programm druckt das Wort 'YES' aus, wenn Sie 'Y' eingeben, und druckt das Wort 'NO' wenn Sie etwas anderes eingeben.

GPRINT C,V,Gs

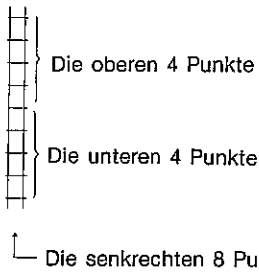
FORMAT: 1. GPRINT zeichenfolge
2. GPRINT ausdruck; ausdruck; ausdruck;...
3. GPRINT

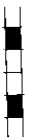
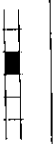
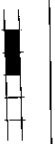
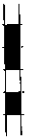
Abkürzung: GP.
Vergleiche: GCURSOR, PRINT

Wirkung:
Wirkung Gibt das angegebene Punktmuster aus.

Anmerkungen:
Das GPRINT-Kommandos gibt das angegebene Punktmuster aus. Ein Punktmuster besteht aus einer senkrechten Linie von 8 Punkten.

Im Format (1) wird das 8-Punktmuster in 4 untere und 4 obere Punkte durch eine in Anführungszeichen (" ") gegebenen Zeichenfolge geteilt, wobei die 4-Punktmuster als hexadezimale Zahlen wiedergegeben werden.

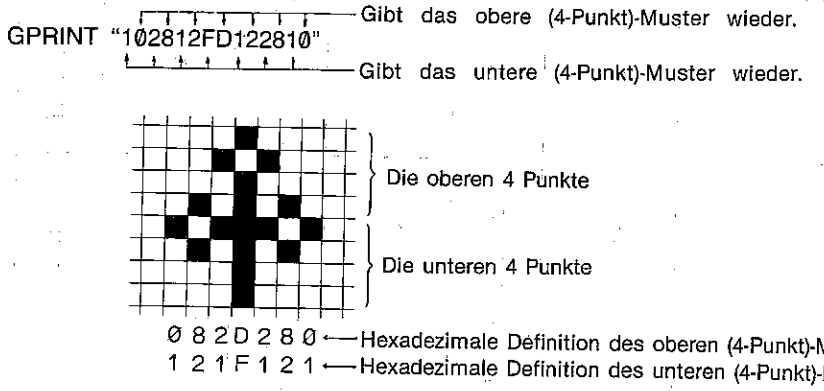


Hexadezimale Zahl	Punkmuster	Hexadezimale Zahl	Punkmuster	Hexadezimale Zahl	Punkmuster	Hexadezimale Zahl	Punkmuster
0		4		8		C	
1		5		9		D	
2		6		A		E	
3		7		B		F	

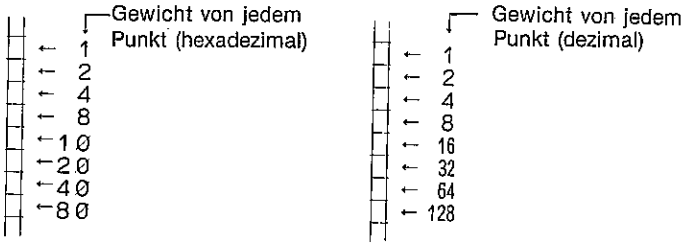
GPRINT "○○ ○○ ○○ ..."

Eine Einzelgruppe (Paar) von 2 numerischen Zeichen gibt ein senkrechtes (8-Punkt)-Muster an. Jedes numerische Zeichen, das am Ende der Zeichenfolge übrigbleibt und kein Paar bildet, wird ignoriert.

Beispiel:


GPRINT "102812FD122810"
Gibt das obere (4-Punkt)-Muster wieder.
Gibt das untere (4-Punkt)-Muster wieder.
Die oberen 4 Punkte
Die unteren 4 Punkte
Ø 8 2 D 2 8 Ø ← Hexadezimale Definition des oberen (4-Punkt)-Musters
1 2 1 F 1 2 1 ← Hexadezimale Definition des unteren (4-Punkt)-Musters

Im Format (2) wird ein senkrechten 8-Punktmuster als Hexadezimal- oder Dezimalwert wiedergegeben. Ein "Gewicht" wird jedem Punkt in dem senkrechten 8-Punktmuster zugewiesen wie unten angegeben.



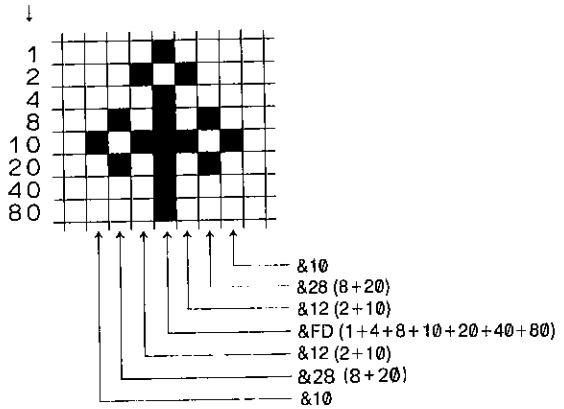
Geben Sie das Punktmuster mit einem numerischen Wert an, mit der Summe der Gewichte der Punkte, die auf dem Bildschirm zu setzen sind.

Beispiel:

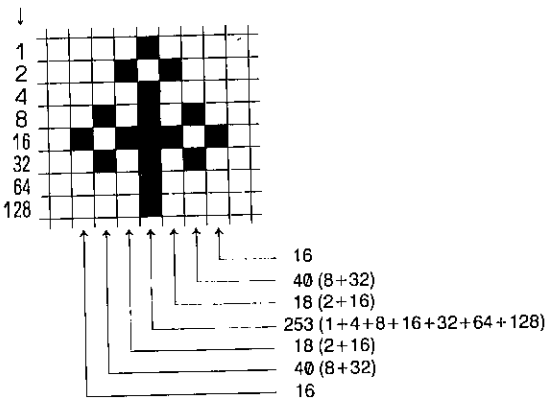
Hexadezimale Definition von Punktmustern.
GPRINT &1Ø; &28; &12; &FD; &12; &2B; &1Ø

Dezimale Definition von Punktmustern.
GPRINT 16; 4Ø; 18; 253; 18; 4Ø; 16

Hexadezimales Gewicht



Dezimales Gewicht

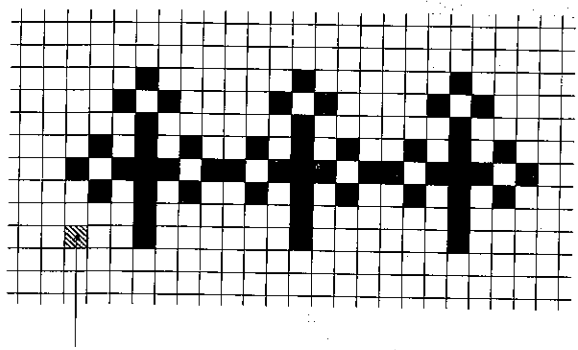


Im Format (3) wird die vorherige Grafik-Anzeige weiterhin dargestellt.

Das GCURSOR-Kommando wird benutzt, um den Cursor auf dem Bildschirm zu setzen, bevor Grafiken mit GPRINT auf den Bildschirm übertragen werden. Bei der Ausgabe des Grafik-Musters mit GPRINT befindet sich der Cursor in der linken unteren Ecke, d.h. beim ersten Punkt von der untersten Zeile des ersten senkrechten 8-Punktmusters, das angegeben wird.

Beispiel:

```
10: AA$ = "102812FD122810"
20: GCURSOR (60, 20)
30: GPRINT AA$;AA$;AA$
```



Die mit dem GCURSOR-Kommando (6Ø, 2Ø) angegebene Position. Die Anzeige des Musters beginnt an dieser Position mit Hilfe der oberen 8 Punkte.

Endet GPRINT mit einem Semikolon ";", so wird der Cursor waagrecht zum Punkt bewegt, beginnend in der Spalte, die direkt rechts vom eben angezeigten Muster liegt. Wenn GPRINT mit einem Doppelpunkt ":" oder nichts endet, wird die Position des Cursors nicht aktualisiert und das folgende GPRINT-Kommando wird ausgegeben, beginnend an der gleichen Stelle.

GRAD

C,V

FORMAT: 1. GRAD

Abkürzung: GR.
Vergleiche: DEGREE, RADIAN

Wirkung:
GRAD wird benutzt, um die Form von Winkelwerten auf Neu-Grad umzuschalten.

Anmerkungen:
Der PC-1360 kann drei Formen der Winkeldarstellung verarbeiten: Alt-Grad (dezimal), Radial-Werte und Neu-Grad. Diese Formen werden für die Darstellung der Werte der Argumente von den SINus-, COSinus- und TANgens-Funktionen und der Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens benutzt.
Die GRAD-Funktion wird benutzt, um Winkelwerte in Gradienten umzuformen, bis DEGREE oder RADIAN benutzt wird. Die Neu-Grad-Form stellt die Winkelmessung in Form von Prozentgradient dar, d.h. ein Winkel von 45° ist ein Gradient von 5Ø°.

Beispiele:
1Ø: GRAD
2Ø: X = ASN 1
3Ø: PRINT X

X hat jetzt den Wert 1ØØ, d.h. 1ØØ Grad, dem Arcussinus von 1,.

GRAPH C,V,Gp

FORMAT: 1. GRAPH

Abkürzung: GRAP.
Vergleiche: LTEXT

Wirkung:
GRAPH wird benutzt, um den Drucker in den Grafik-Modus zu setzten.

Anmerkungen:
Wenn Sie diesen Befehl ausführen, wird der Drucker vom Text-Modus in den Grafik-Modus gesetzt, um eine grafische Darstellung zu zeichnen.

Nach der Ausführung des LLIST-Kommandos oder nach dem Drucken im manuellen Betrieb mit angeschlossenem CE-140P wird der Drucker automatisch wieder vom Grafik-Modus in den Text-Modus umgeschaltet.

Um Zeichen im Grafik-Modus auszudrucken, benutzen Sie den LPRINT-Befehl in einem der beiden Formate:

(1) LPRINT "P zeichenfolge"
(2) LPRINT "P" $\left\{ \begin{matrix} + \\ ; \end{matrix} \right\}$ zeichenfolgen-variable

1. Wenn Sie den Betrieb des Druckers durch Betätigen der **BRK**-Taste unterbrechen, während der Drucker eine Figur im Grafik-Modus zeichnet, achten Sie darauf folgendes einzugeben:
   LPRINT" "**ENTER**
   Sonst kann es sein, daß die nachfolgenden Kommandos an den Drucker nicht richtig ausgeführt werden.

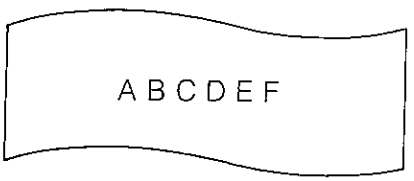
2. Um den Druckkopf an die äußerste linke Position zurückzubringen, geben Sie folgendes ein:
   LTEXT **ENTER**
   LPRINT **ENTER**
   In diesem Fall wird aber der Drucker aus dem Grafik-Modus freigegeben.

3. Wenn Sie irgend ein druckerbezogenes Kommando im Text-Modus benutzten, das nur im Grafik-Modus gültig ist (wie z.B. CIRCLE, CROTATE, GLCURSOR usw.), druckt der Drucker Zeichen entsprechend dem Anweisungszeichen aus.

Beispiel:

```
 5: OPEN
1Ø: GRAPH
2Ø: GLCURSOR (2ØØ, −3Ø)
3Ø: LPRINT "PABCDEF"
4Ø: LTEXT
5Ø: LPRINT
6Ø: END
```

[5] Dieses Kommando wird für den CE-515P benötigt.



IF ... THEN V

FORMAT: 1. IF bedingung THEN anweisung
2. IF bedingung anweisung

Abkürzung: IF T.,
Vergleiche:

Wirkung:
IF...THEN wird benutzt, um eine Anweisung auszuführen oder nicht auszuführen, wenn bestimmte Bedingungen bei der Programmausführung erfüllt sind.

Anmerkungen:
Im normalen Ablauf von BASIC-Programmen werden die Anweisungen in der Reihenfolge ihres Auftretens abgearbeitet. Das IF...THEN-Befehlspaar erlaubt es, eine bestimmte Anweisung nur dann auszuführen, wenn eine bestimmte Bedingung erfüllt ist. Ist der Bedingungsteil der IF-Anweisung wahr, sor wird die Anweisung ausgeführt, ist die Bedingung falsch, wird sie übersprungen.

Der Bedingungsteil der IF-Anweisung kann jeder vergleichende Ausdruck, wie in Kapitel 4 beschrieben, sein. Es ist auch möglich, einen numerischen Ausdruck als Bedingung zu setzen, obgleich der Sinn der Anweisung dadurch wenig klar wird. Jeder Ausdruck, der kleiner oder gleich Null ist, setzt die Bedingung falsch. Jeder positive Wert setzt die Bedingung wahr.

Die Anweisung, die auf den THEN-Teil folgt, kann jede BASIC-Anweisung sein, auch eine weitere IF....THEN-Anweisung. Handelt es sich um eine LET-Anweisung, so muß der LET-Befehl angegeben werden.

Die beiden Formen der IF-Anweisung sind in der Ausführung gleich, aber die erste Form ist wesentlich klarer.

Beispiele:

```
1Ø: INPUT "CONTINUE?"; A$
2Ø: IF A$="YES" THEN GOTO 1Ø
3Ø: IF A$="NO" THEN GOTO 6Ø
4Ø: PRINT "YES OR NO, PLEASE"
5Ø: GOTO 1Ø
6Ø: END
```

Dieses Programm fragt andauernd 'CONTINUE?', solange 'YES' eingegeben wird; es stoppt, wenn 'NO' eingegeben wird, und beanstandet andernfalls.

INKEY$ V,F

FORMAT: 1. INKEY$

Abkürzung: INK.

Vergleiche:

Wirkung:
INKEY$ ist eine Zeichenfolgen-Pseudovariable, die der angegebenen Variable den Wert der betätigten Taste zuweist, während die INKEY$-Funktion ausgeführt wird.

Anmerkungen:
INKEY$ wird benutzt, um auf die Betätigung einzelner Tasten zu antworten, ohne auf das Ende der Eingabe durch die **ENTER** -Taste zu warten.

Die Codewerte der Tasten finden Sie in der Tastencode-Tabelle im Anhang.

Beispiel:
```
1Ø: A$ =  INKEY$
2Ø: B =  ASC A$
3Ø: IF B = Ø THEN GOTO 1Ø
4Ø: IF B...
```

Die Zeilen 4Ø und weiter enhalten Tests für die Taste und die nötigen Bedienungsmaßnahmen. (Zum Beispiel: 4Ø: PRINT A$)

INPUT V

FORMAT: 1. INPUT eingabe liste

Wo: eingabe liste	ist eingabe gruppe oder eingabe gruppe, eingabe liste
und: eingabe gruppe	ist var liste oder: aufforderungszeichen, var liste oder: aufforderungszeichen; var liste
und: var liste	ist variable oder variable, var liste
und: aufforderungszeichen	ist jede Zeichenfolgen-Konstante

Abkürzung: I.
Vergleiche: INPUT #, READ, CURSOR, PRINT

Wirkung:
INPUT wird benutzt, um einen oder mehrere Werte über die Tastatur einzugeben.

Anmerkungen:
Wollen Sie verschiedene Werte für jeden Programmlauf änern, benutzen Sie INPUT, um diese Werte über die Tastatur einzugeben.

In der einfachsten Form hat die INPUT-Anweisung keine Aufforderungszeichenfolge, sondern ein Fragezeichen erscheint in der linken oberen Ecke der Anzeige. Sie geben dann einen Wert, gefolgt von **ENTER**, ein. Dieser Wert wird dann der ersten Variablen der Variablenliste zugewiesen. Sind mehrere Variablen in der gleichen INPUT-Anweisung enthalten, so wird dieser Prozeß so oft wiederholt, bis die Liste erschöpft ist.

Wenn ein Aufforderungszeichen in der INPUT-Anweisung enthalten ist, so ist der Prozeß genau derselbe, nur daß, anstelle eines Fragezeichens, ein Aufforderungszeichen in der linken oberen Ecke der Anzeige erscheint.

Wenn auf die Aufforderungszeichenfolge ein Semikolon folgt, erscheint der Cursor unmittelbar nach dem Aufforderungszeichen. Wenn auf das Aufforderungszeichen ein Komma folgt, erscheint das Aufforderungszeichen. Wenn dann eine Taste betätigt wird, wird die Anzeige gelöscht und das erste Zeichen der Eingabe erscheint am linken Rand.

Wenn ein Aufforderungszeichen angegeben wird und die darauf folgende Variablenliste mehr als eine Variable enthält, so werden die zweite und die folgenden Variablen mit einem Fragezeichen gekennzeichnet. Beinhaltet die Liste ein zweites Aufforderungszeichen, so wird es für die unmittelbar darauf folgende Variable angezeigt.

Wurde die Startposition des Bildschirms mit Hilfe des CURSOR-Kommandos vor der Ausführung des INPUT-Kommandos angegeben, so erscheint ein Aufforderungszeichen oder "?" an dieser Stelle.

Wenn Sie beim Betätigen der ENTER-Taste keinen Wert eingeben, so behält die Variable den Wert, den sie vor der INPUT-Anweisung hatte.

Wenn Sie bei der Eingabe Ihres Wertes einen Fehler machen, drücken Sie die CLS-Taste, um die Eingabe zu löschen und bringen Sie den Cursor wieder zurück an die Startposition. Sie können auch die ▶- und ◀-Tasten zur Positionierung des Cursors benutzen, aber diese löschen nicht zuerst die falsche Eingabe, was ihren Gebrauch schwieriger macht.

Beispiele:

```
10: INPUT A
20: INPUT "A=";A
30: INPUT "A=",A
40: INPUT "X=?";X,"Y=?";Y
```

[10] Löscht das Anzeigefeld und setzt ein Fragezeichen an den linken Rand.
[20] Gibt 'A=' aus und wartet auf eine Eingabe.
[30] Gibt 'A=' aus.
Bei der Eingabe verschwindet 'A=' und der eingegebene Wert wird am linken Rand angezeigt.
[40] "X=?"; Gibt 'X=?' aus und wartet auf die erste Eingabe. Nach dem ENTER wird die Anzeige gelöscht und 'Y=?' wird am linken Rand angezeigt.

INPUT

FORMAT: 1. INPUT # var liste
2. INPUT # "dateiname"; var liste

Wo: var liste — ist variable
oder variable, var liste

Abkürzung: I.#
Vergleiche: INPUT, PRINT #, READ

Wirkung:
INPUT# wird benutzt, um Daten von einer Kassette einzugeben.

Anmerkungen und Beispiele:
Folgende Variablentypen können in der INPUT#-Anweisung eingesetzt werden:
(1) Festpunktvariablen—A, B C, A(7), D*, A(2Ø)*, usw.
(2) Einfache Variablen—AA, B3, CP$, usw.
(3) Feldvariablen—S(*), HP(*), K$(*), usw.

1) Übergabe von Daten auf Festpunktvariablen
Um Daten von der Kassette auf Festpunktvariablen zu übertragen, müssen die Variablennamen in der INPUT#-Anweisung angegeben werden.

```
INPUT # "DATA 1";A,B,X,Y
```

Diese Anweisung weist die Daten der Banddatei namens "DATA 1" den Variablen A, B, X und Y in der gegebenen Reihenfolge zu.

Um alle verfügbaren Festpunktvariablen und, falls definiert, erweiterte Variablen (A(27) und weiter) mit Daten von der Kassette zu belegen, muß der ersten Variablen ein Stern (*) angehängt werden.

```
INPUT # "D-2";D*
```

Mit dieser Anweisung wird der Inhalt der Banddatei "D-2" den Variablen D bis Z und A(27) und höher zugewiesen.

INPUT # A(1Ø)✱ (ohne DIMensionierung)

Diese Anweisung überträgt die Daten der ersten gefundenen Datei nach dem Bandanfang von der Kassette auf die Variablen A(1Ø) und weiter (also J bis Z und A(27) und weiter).

Wurde bereits mit einem DIM-Befehl ein Feld namens A definiert, können keine indizierten Festpunktvariablen der Form A( ) definiert werden.

Die Übertragung von Daten auf Festpunktvariablen und erweiterte Variablen (A(27) und weiter) wird so lange fortgesetzt, bis das Ende der Ursprungsdatei auf dem Band gefunden wird, aber wenn der Speicher des Computers voll ist, erhalten Sie eine Fehlermeldung (ERROR 6).

2) Datenübertragung auf einfache Variablen

Daten in einer Kassettendatei können auf einfache Variablen übertragen werden, indem die gewünschten einfachen Variablen in die INPUT#-Anweisung eingegeben werden.

INPUT # "DM-1";AB, Y1, XY$

Diese Anweisung überträgt die Daten von der Banddatei namens "DM-1" auf die einfachen Variablen AB, Y1 und XY$.

Numerische Daten müssen auf numerische einfache Variablen übertragen werden, und Zeichendaten müssen einfache Zeichenvariablen sein. Andere Übertragungen sind nicht möglich.

Im Programm-Datenspeicherbereich muß Raum für die einfachen Variablen frei gehalten werden, bevor die INPUT#-Anweisung ausgeführt werden kann. Sonst erhalten Sie eine Fehlermeldung. Verwenden Sie Platzreservierungs-Anweisungen, um Plätze für die einfachen Variablen zu reservieren.

AA = Ø **ENTER**
B1$ = "A" **ENTER**
INPUT # AA, B1$ **ENTER**

Verwenden Sie numerische Werte oder Zeichen in den Reservierungs-Anweisungen, um Plätze für die Variablen zu reservieren.

3) Datenübertragung auf Feldvariablen

Um Daten von einer Banddatei auf Feldvariablen zu übertragen, muß der Name des Feldes in der INPUT#-Anweisung in der Form feldname(*) angegeben werden.

```
5Ø DIM B(5)
6Ø INPUT # "DS-4";B(*)
```

Diese Anweisung überträgt die Daten von der Banddatei namens "DS-4" auf die Variablen (B(Ø) bis B(5)) im Feld B.

Numerische Daten müssen auf numerische Feldvariablen gleicher Länge wie die der Daten, Zeichendaten müssen auf Zeichenfeldvariablen gleicher Länge wie die der Daten übertragen werden. Wird dies nicht beachtet, erhalten Sie eine Fehlermeldung.

Sie müssen Raum im Programm-Datenspeicherbereich für die Feldvariablen frei halten, bevor die INPUT#-Anweisung ausgeführt wird. Sonst erhalten Sie eine Fehlermeldung. Benutzen Sie die DIM-Anweisung, um das Feld im voraus zu definieren.

—WICHTIG—

Stimmt die Anzahl der Variablen in der INPUT#-Anweisung nicht mit der Zahl der Daten auf der Kassette überein, geschieht folgendes:

* Ist die Zahl der Daten in der Banddatei (die zu übertragen sind) größer als die Zahl der Variablen, so wird die Datenübertragung bis zur letzten Variable durchgeführt, und die restlichen Daten werden ignoriert.

* Ist die Zahl der Daten in der Banddatei (die zu übertragen sind) kleiner als die Zahl der Variablen, so werden alle Daten der Datei den Variablen bis an das Ende der Datei zugewiesen und die überzähligen Variablen behalten ihren alten Inhalt. In diesem Fall aber wartet der Computer auf die Datenübertragung von der Kassette. Um diesen Zustand zu stoppen, sollten Sie die **BRK**/ON-Taste betätigen.

* Wird die INPUT#-Anweisung ohne Variable eingegeben, erhalten Sie eine Fehlermeldung (ERROR 1).

INPUT#1 C,V,S

FORMAT: INPUT#1 variable, variable, variable...

Abkürzung: I.#1
Vergleiche: OPEN, PRINT#1

Wirkung:
Lädt Daten über das serielle E/A-Interface (Terminal) und weist sie definierten Variablen zu.

Anmerkungen:
Dieses Kommando wird nur ausgeführt, wenn die Leitung zum seriellen E/A-Interface geöffnet wurde (nach der Ausführung von OPEN) und wird ansonsten ignoriert.

INPUT#1 weist den Variablen Daten in gleicher Weise wie das PRINT#1-Kommando zu. Für weitere Einzelheiten siehe das PRINT#1-Kommando.

Beispiel:
INPUT#1A, AB, C$, E(*)

Die empfangenen Daten werden den Variablen A, AB und C$ und der Feldvariablen E( ) über das serielle E/A-Interface zugewiesen.

Achten Sie darauf, daß die definierten Variablen mit den eingegebenen Daten übereinstimmen (z.B. Zeichentypen oder numerische Typen).

Wird im ASCII-Codesystem ein Zeichen einer numerischen Variablen zugewiesen, so wird ihr Wert gleich Ø gesetzt. Wird eine Zahl einer Zeichenvariablen zugewiesen, so wird der Inhalt in eine Zeichenfolge umgewandelt. Sind die Variablen und Daten nicht gleichen Typs, werden Sie also ungewollte Werte führen.

Zum Beispiel, wird "SIN 3Ø" einer numerischen Variablen zugewiesen, wird Ø in der Variablen gespeichert, und "1Ø + 4Ø" wird nur als 1Ø gespeichert, da alles, was nach dem Aufforderungszeichen folgt, ignoriert wird. Wenn Sie CR (Kontrollcode &ØD) oder NULL (&ØØ) benutzen, werden alle folgenden eingegebenen Daten ignoriert.

Einfache und Feldvariablen müssen im Programm/ Datenspeicherbereich Platz zugewiesen bekommen, bevor das INPUT # 1-Kommando ausgeführt wird. Wenn diese Variablen nicht zugewiesen werden, erhalten Sie eine Fehlermeldung.

INT C,V,F

FORMAT: 1. INT(X)

Abkürzung:
Vergleiche:

Wirkung:
Schneidet den Dezimalteil einer reellen Zahl ab und errechnet eine ganze Zahl.

Anmerkungen:
Der Wert X wird auf eine ganze Zahl abgerundet. Das ganzzahlige Ergebnis ist immer kleiner oder gleich X, ganz gleich ob X positiv oder negativ ist. Bei negativen Zahlen wird das absolute ganzzahlige Ergebnis also größer oder gleich groß wie der absolute Wert sein.

Beispiel:

```
 5: WAIT 60
10: A=-3.3: PRINT A,  INT (A)
20: B=-1.9: PRINT B,  INT (B)
30: C=-0.5: PRINT C,  INT (C)
40: D=0.2:  PRINT D,  INT (D)
50: E=1.6:  PRINT E,  INT (E)
60: F=3:    PRINT F,  INT (F)

RUN
          -3.3        -4.
          -1.9        -2.
          -0.5        -1.
           0.2         0.
           1.6         1.
            3.          3.
```

LEFT$

C,V,F

FORMAT: 1. LEFT$(X$,N)
2. LEFT$("string", N)

Abkürzung: LEF.

Vergleiche: MID$, RIGHT$

Wirkung:
Nimmt die Zeichen N vom linken Ende einer Zeichenfolge X$ heraus.

Amerkungen:
Der Wert von N muß zwischen 0 und 80 liegen. Bruchteile werden abgerundet (abgeschnitten). Ist N kleiner als 1, wird eine Nullzeichenfolge herausgenommen. Ist N größer als die Anzahl der Zeichen in X$, wird die ganze Zeichenfolge herausgenommen.

Beispiel:

```
1Ø: X$ = "SHARP"
2Ø: FOR N = 1 TO 6
3Ø: LET S$ = LEFT$ (X$,N)
4Ø: PRINT S$
5Ø: NEXT N

RUN
S
SH
SHA
SHAR
SHARP
SHARP
```

LEN

C,V,F

FORMAT: 1. LEN(X$)
2. LEN ("zeichenfolge")

Abkürzung:
Vergleiche:

Wirkung:
Errechnet die Anzahl der Zeichen in der Zeichenfolge X$.

Anmerkungen:
Die Anzahl der Zeichen in der Zeichenfolge schließt alle Leerzeichen oder Druckzeichen wie Kontrollcodes oder Wagenrückläufe ein.

Beispiel:

```
10: INPUT "ENTER A WORD ";A$
20: N=  LEN A$
30: PRINT "THE WORD LENGTH IS ";N
40: END

RUN
ENTER A WORD CHERRY
THE WORD LENGTH IS 6
```

[10] Gibt ein Wort ein. In diesem Beispiel gibt der Verwender "CHERRY" ein.
[20] Ermittelt die Länge eines Wortes.
[30] Druckt die Antwort aus.

LET

V

FORMAT: 1. LET variable = ausdruck
2. variable = ausdruck

Abkürzung: LE.
Vergleiche: IF ~ THEN

Wirkung:
LET wird benutzt, um einer Variablen einen Wert zuzuweisen.

Anmerkungen:
Mit LET weisen Sie der angegebenen Variable den Wert des Ausdrucks zu. Die Form des Ausdrucks muß mit der Variablen übereinstimmen, d.h. nur numerische Ausdrücke können numerischen Variablen und nur Zeichenfolgen-Ausdrücke können Zeichenfolgen-Variablen zugewiesen werden. Um Werte aus der einen in die andere Form zu übertragen, muß eine der beiden expliziten Umsetzfunktionen STR$ oder VAL eingesetzt werden.

Der LET-Befehl kann in allen LET-Anweisungen weggelassen werden, es sei denn, die Anweisungen folgen auf das THEN einer IF...THEN-Anweisung.

Beispiele:

```
10: I = 10
20: A=5*I
30: X$ = STR$(A)
40: IF I>=10 THEN LET Y$=X$+".00"
```

[10] Weist I den Wert 10 zu.
[20] Weist A den Wert 50 zu.
[30] Weist X$ den Wert 50 zu.
[40]Weist Y$ den Wert 50.00 zu.

LF C,V,Gp

FORMAT: 1. LF
2. LF ausdruck

Abkürzung:
Vergleiche:

Wirkung:
LF wird für den Druckpapiervorschub benutzt.

Anmerkungen:
Dieser Befehl ist nur im Text-Modus gültig.

Mit dem Format 1 läßt der Drucker das Papier um eine Zeile vorschieben.

Mit dem Format 2 läßt der Drucker das Papier um die definierte Anzahl von Zeilen vorschieben. Der Wert von ausdruck muß zwischen −999 und 999 liegen.

Ist der Wert von ausdruck positiv, läuft das Papier vorwärts ein. Ist der Wert negativ, läuft das Papier rückwärts ein. Bei der Ausführung des LF-Befehls ist der Zeilenabstand der gleiche wie bei CSIZE.

Beispiel:
LF 1Ø

Schiebt das Papier um 10 Zeilen vor.

LINE C,V,Gs

FORMAT:

1. LINE (ausdruck 1, ausdruck 2)—(ausdruck 3, ausdruck 4), $\left\{ \begin{matrix} S \\ R \\ X \end{matrix} \right\}$, expression 5, $\left\{ \begin{matrix} B \\ BF \end{matrix} \right\}$

(A) (B) (C) (D) (E)

Abkürzung: LIN.
Vergleiche: GCURSOR, PSET

Wirkung:
Zieht eine Linie zwischen 2 definierten Punkten.

Anmerkungen:
Zwischen den 2 durch (ausdruck 1, ausdruck 2) und (ausdruck 3, ausdruck 4) definierten Punkten wird eine Linie gezogen.

Beispiel:
LINE (Ø, Ø)—(149,31)

Zieht eine Linie von der oberen linken zur unteren rechten Ecke des Anzeigefeldes (Bildschirm).

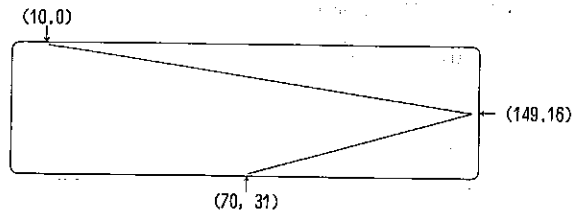
Die Werte von ausdruck 1 bis ausdruck 4 in Term (A) und (B) können im Bereich von −32768 bis +32767 liegen, obwohl, um im Bildschirm dargestellt werden zu können, ausdruck 1 und ausdruck 3 zwischen Ø und 149, und ausdruck 2 und ausdruck 4 zwischen Ø und 31 liegen müssen.

Sie erhalten keine Fehlermeldungen, wenn die Werte innerhalb der Maschinengrenzen von −32768 bis 32767 liegen, auch wenn nur die Teile der Linie angezeigt werden, die den Bildschirm durchkreuzen. Liegen die Werte außerhalb der Maschinengrenzen, erhalten Sie die Fehlermeldung ERROR 3.

Term (A) (ausdruck 1, ausdruck 2) kann weggelassen werden. Wird er weggelassen, wird die Linie vom Punkt (Ø, Ø) oder dem durch Term (B) (ausdruck 3, ausdruck 4) definierten Punkt mit Hilfe eines direkt vorher gegebenen LINE-Kommandos gezogen.

Beispiel:

```
 5: CLS : WAIT Ø
1Ø: LINE (1Ø, Ø)-(149, 16)
2Ø: WAIT:LINE-(7Ø, 31)
```



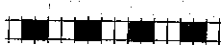
Da jede Linie aus einzelnen Punkten auf dem Bildschirm aufgebaut ist, kann es sein, daß Diagonalen nicht ganz gerade erscheinen.

Term (C) gibt den Punkttyp an:

S: Zieht die Linie mit den gesetzten Punkten. (Setzt die Punkte)
R: Zieht die Linie mit den gelöschten Punkten. Diese Möglichkeit wird benutzt, um eine Linie in einem Bereich zu ziehen, wo die umgebenden Punkte hell sind oder um eine bestehende Linie zu löschen. (Setzt die Punkte zurück)
X: Zieht die Linie und löscht die Punkte, wenn diese schon hell sind oder setzt die Punkte, wenn diese noch nicht hell sind. (Kehrt die Punkte um)
Wenn Sie weder S, R noch X angeben, wird S angenommen.

Term (D) gibt den Typ der Linie an, ob durchgehend oder gestrichelt, indem er eine Bitmustersequenz definiert.

Wenn der Wert von ausdruck 5 zum Beispiel gleich 26214 (&6666) ist, so wird eine Linie mit folgendem Muster gezogen.



16 Punkte — Die Linie wird gebildet, indem das links abgebildete Muster wiederholt wird.

Die Zahl 26214 (&6666) kann als Binärzahl wiedergegeben werden.

Ø11ØØ11ØØ11ØØ11Ø

Wenn Sie die 16 Punkte der Linie in der obigen Abbildung mit der Binärzahl vergleichen, werden Sie erkennen, daß die gesetzten Punkte den Einsen entsprechen und daß die ungesetzten Punkte, die den Nullen entsprechen, gelöscht werden. Das Muster der Linie wird also durch die Nullen und Einsen wiedergegeben, nachdem der Wert von ausdruck 5 in eine 16stellige Binärzahl umgesetzt wurde. Die Linie erscheint also nicht auf dem Bildschirm, wenn der Wert von ausdruck 5 gleich Ø ist, und eine durchgezogene Linie erscheint, wenn der Wert 65535(&FFFF)ist. Wenn ausdruck 5 weggelassen wird, erscheint auch eine durchgezogene Linie. Wenn Sie aber in Term (C) R angeben, geschieht das Gegenteil, und wenn Sie X angeben, werden die den Einsen entsprechenden Punkte umgekehrt.

Der Wert von ausdruck 5 kann im Bereich zwischen Ø und 65535(&FFFF) liegen.

Term (E) zieht ein Quadrat, dessen Diagonale eine Linie ist, die 2 durch Term (A) und Term (B) definierte Punkte verbindet.

B: Zieht ein Quadrat.
BF: Zieht ein Quadrat, das mit Linien gefüllt ist.

Beispiele:

```
1Ø: CLS : WAIT Ø
2Ø: AA$ = "1Ø2812FD122812Ø"
3Ø: GCURSOR (64, 2Ø)
4Ø: GPRINT AA$; AA$; AA$
5Ø: LINE (24, Ø) - (124, 31), &F18F, B
6Ø: LINE (34, 3) - (114, 28), X, BF
7Ø: GOTO 6Ø
```

LIST C

FORMAT: 1. LIST
2. LIST zeilennummer
3. LIST "etikett"

Abkürzung: L.
Vergleiche: LLIST

Wirkung:
Wirkung Mit dem LIST-Kommando wird ein Programm zur Anzeige gebracht.

Anmerkungen:
Das LIST-Kommando kann nur im PROgramm-Modus benutzt werden.

Mit dem Format (1) wird das Programm von der ersten Zeile angezeigt, so weit, bis das Display voll ist. Mit dem Format (2) wird das Programm von der angegebenen Zeilennummer angezeigt, so weit, bis das Display voll ist.

Gibt es die anegegebene Zeilennummer nicht, so wird das Programm ab der Zeile angezeigt, deren Zeilennummer der angegebenen am nächsten kommt.

Mit dem Format (3) wird das Programm ab der mit diesem Etikett versehenen Zeile angezeigt, so weit, bis das Display voll ist.

Sind mit dem MERGE-Kommando mehrere Programme in den Speicher geladen, so wird mit dem LIST-Kommando das letzte eingeladene Programm gelistet.

Wird aber das im Format (3) angegebene Etikett im zuletzt geladenen Programm nicht gefunden, so sucht das Kommando das Etikett vom ersten Programm an ab. Wird das angegebene Etikett gefunden, so wird die dazugehörige Zeile angezeigt. Falls ein Passwort angegeben wurde, wird das LIST-Kommande ignoriert.

Beispiele:

LIST 1ØØ

Zeigt die Zeilennummer 1ØØ an.

LLINE

C,V,Gp

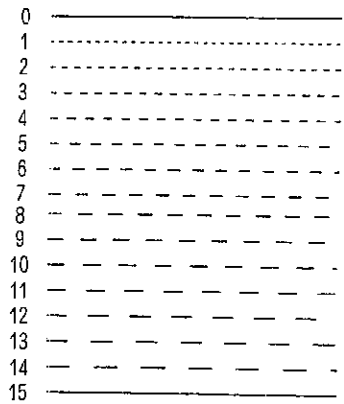
FORMAT: 1. LLINE [(ausdruck 1, ausdruck 2)]—(ausdruck 3, ausdruck 4) [,ausdruck 5][, ausdruck 6][,B]

Abkürzung: LLIN.
Vergleiche: RLINE, PAINT, COLOR

Wirkung:
Der LLINE-Befehl wird benutzt, um eine Linie zwischen zwei definierten Punkten zu ziehen.

Anmerkungen:
Dieser Befehl ist nur gültig im Grafik-Modus und wird benutzt, um eine Linie zwischen dem durch (ausdruck 1, ausdruck 2) definierten Koordinatenpunkt und dem durch (ausdruck 3, ausdruck 4) definierten Koordinatenpunkt zu ziehen. (ausdruck 1, ausdruck 2) kann weggelassen werden. Wird er weggelassen, wird eine Linie vom jetzigen Punkt des Zeichenstifts bis zu dem durch (ausdruck 3, ausdruck 4) definierten Punkt gezogen.

Ausdruck 5 wird benutzt, um einen der folgenden Linientypen anzugeben, indem dem ausdruck ein Wert (Ø bis 15) zugewiesen wird. Der Vorgabewert von ausdruck 5 ist Ø.



Ausdruck 6 wird benutzt, um die Farbe der zu ziehenden Linie anzugeben. Der Wert von ausdruck 6 kann im erweiterten Farb-Modus zwischen 0 und 7 und im normalen Farb-Modus zwischen Ø und 3 liegen. (Was die durch jeden Wert definierte Farbe betrifft, siehe den COLOR-Befehl.)

Ausdruck 6 kann weggelassen werden. Wird er weggelassen, wird der vorherige Wert angenommen.

Diese Figur wird in Wirklichkeit in Farbe ausgedruckt. Wenn Sie B am Ende des LLINE-Befehls eingeben, wird ein Rechteck mit Hilfe des durch (ausdruck 1, ausdruck 2) definierten Koordinatenpunktes und des durch (ausdruck 3, ausdruck 4) definierten Koordinatenpunktes als Endpunkte der Diagonale gezogen.

Beispiel: LLINE (1Ø, 2Ø)–(2ØØ, –2Ø), 2,Ø,B

Achtung:
(ausdruck 1, ausdruck 2) kann nicht weggelassen werden, wenn Sie ein rechteck ziehen wollen.

Werden nach der Ausführung dieser Befehle mit B die Befehle LLINE oder RLINE ohne B ausgeführt, sollte der nachfolende Befehl unmittelbar nach der Ausführung dieser Befehle ohne B ausgeführt werden.

```
POKE &FB2Ø,Ø
```

Wird dieser Befehl nicht ausgeführt, mag der Computer ein Rechteck mit LLINE oder RLINE ohne B zeichnen.

Befehlsspezifikation für durchgezogene Linien
Der LLINE-Befehl in den folgenden Formaten erlaubt dem Drucker, durchgezogene Linien zu zeichnen.

LLINE (ausdruck 1, ausdruck 2)–(ausdruck 3, ausdruck 4)–(ausdruck 3, ausdruck 4)–(ausdruck 3, ausdruck 4)–(ausdruck 3, audruck 4)–(ausdruck 3, ausdruck 4), ausdruck 5, ausdruck 6

Im obigen Format kann eine durchgehende Verbindungaline von dem durch (Ausdruck 1, Ausdruck 2) definierten Punkt zu dem durch (Ausdruck 3, Ausdruck 4) definierten Punkt, eine weitere von dem durch (Ausdruck 3, Ausdruck 4) definierten Punkt zu dem durch das nächsten Paar definierten Punkt ..... gezeichnet werden. Es können maximal sechs Paare angegeben werden.

Beispiele:

```
  5: OPEN
 10: GRAPH : RANDOM
 20: GLCURSOR (240, −120)
 25: SORGN
 30: FOR J=0 TO 340 STEP 20
 40: A=107* COS J
 50: B=107* SIN J
 60: R= RND 4−1
 70: LLINE (0, 0)−(A,B), 0,R
 80: NEXT J
 90: LTEXT
100: LPRINT
110: END
```

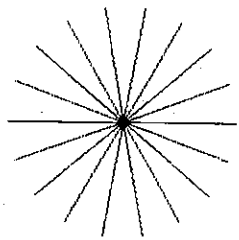
[5] Dieses Kommando wird für den CE-515P benötigt

[25] Bewegt den Zeichenstift bis etwa in die Mitte des Blattes und definiert dort den Ursprung der Koordinaten für das Zeichnen einer Figur.

[60] Eine Zufallszahl von 0 bis 3 wird R zugewiesen.

[70] Die Farbe wird durch den Wert von R angegeben.

[90] Der Drucker wird in den Text-Modus zurückgesetzt und bewegt den Druckkopf bis an die äußerste linke Position.



LLIST AUSGABE ÜBER DAS DRUCKERINTERFACE C,P

Sie können LLIST benutzen, um Programmzeilen direkt an den Drucker über das Druckerinterface auszugeben, oder über das serielle E/A-Interface auszugeben

FORMAT: 1. LLIST
2. LLIST ausdruck
3. LLIST ausdruck 1, ausdruck 2
4. LLIST ausdruck,
5. LLIST,ausdruck

Abkürzung: LL.
Vergleiche: LIST

Wirkung:
Mit dem LLIST-Kommando kann ein Programm auf den wahlweisen Druckern CE-126P oder CE-140P ausgegeben werden.

Anmerkungen:
Beachten Sie, daß der CE-140P zwar an das serielle E/A-Interface angeschlossen ist, sich aber wie an das Drucker-Interface angeschlossen verhält.
Wenn der serielle E/A-Anschluß durch das OPEN-Kommando geöffnet wurde, kann durch das LLIST-Kommando das Programm auf dem an den E/A-Anschluß angeschlossenen Terminal ausgegeben werden. (Siehe Seite ??.)
Um das Programmausgabe-Kommando an den Drucker zurückzugeben, benutzen Sie das CLOSE-Kommando.

Das LLIST-Kommando kann im PROgramm- oder RUN-Modus benutzt werden.

Die erste Form druckt alle im Speicher enthaltenen Programme.

Die zweite Form druckt nur die mit Ausdruck angegebene Zei-lennummer aus.

Die dritte Form druckt die Anweisungen von der Zeilennummer bis zur Zeile, die gleich oder größer ist als der Wert von Ausdruck 1 bis zur nächstgelegenen Zeile, die gleich oder größer ist als der Wert von Ausdruck 2. Zwischen den beiden Nummern müssen mindestens zwei Zeilen liegen.

Die vierte Form listet alle Programmzeilen ab, von der Zeile, deren Nummer durch den Ausdruck angegeben ist.

Die fünfte Form druckt alle Programmzeilen ab bis zur Zeile (inklusive), deren Nummer durch den Ausdruck angegeben ist.

Wenn mit dem MERGE-Kommando mehrere Programme eingeladen wurden, so arbeitet das LLIST-Kommando mit dem zuletzt geladenen Programm. Um ein zuvor geladenes Programm aufzurufen, benutzen Sie das Kommando

LLIST "etikett"

Falls ein Passwort eingegeben wurde, wird das LLIST-Kommando ignoriert.

Beispiele:
LLIST 1ØØ, 2ØØ

Listet die Anweisungen zwischen den Zeilennummern 1ØØ und 2ØØ auf.

LLIST AUSGABE ÜBER DAS SERIELLE E/A-INTERFACE: C,V,S

FORMAT: 1. LLIST

2. LLIST {ausdruck / "etikett"}

3. LLIST ausdruck 1, ausdruck 2

Abkürzung: LL.
Vergleiche: OPEN, CONSOLE

Wirkung:
Sendet die Programminhalte über das serielle E/A-Interface (Terminal) aus.

Anmerkungen:
Das LLIST-Kommando kann während des manuellen Betriebs sowohl im PRO-Modus als auch im RUN-Modus eingesetzt werden. Wenn die Leitung zum seriellen E/A-Interface mit Hilfe des OPEN-Kommandos geöffnet wird, wird das Programm im ASCII-Code gesendet.

Wenn die Leitung geschlossen ist, wird das Programm durch den Drucker ausgedruckt. Das gilt sowohl für den CE-140P als auch für den CE-126P Drucker. Beachten Sie, daß der CE-140P zwar an das serielle E/A-Interface angeschlossen ist, er sich aber wie an das Drucker-Interface angeschlossen verhält.

Im Format (1) werden alle im PC-1360 gespeicherten Programme gesendet.

Wenn das folgende Programm im PC-1360 eingegeben ist, und Sie

LLIST **ENTER**

drücken, wird das Programm im folgenden Format gesendet.

```
  10: OPEN
 100: REM   **ABC-12**
65279: END
```

Space	1	0	:	O	P	E	N	Space	CR
1	0	0	:	R	E	M	Space	Space	*
*	A	B	C	-	1	2	*	*	CR
6	5	2	7	9	:	E	N	D	Space
CR									

Achtung:
CR ist ein Endcode. Er ist entweder LF oder CR + LF je nachdem wie das OPEN-Kommando gesetzt wurde.

Im Format (2) wird die Zeile gesendet, die mit dem Wert von ausdruck oder mit dem eingegebenen Etikett gekennzeichnet ist.

Im Format (3) wird das Programm gesendet, beginnend mit der Zeile, die durch den Wert von ausdruck 1 definiert ist, bis zur Zeile, die durch den Wert von ausdruck 2 definiert ist. (Sie können auch Etikette für ausdruck 1 und ausdruck 2 verwenden.)

Ausdruck 1 oder ausdruck 2 kann im Format (3) weggelassen werden.

Wenn Sie ausdruck 1 weglassen, wird das Programm von der ersten Zeile bis zur Zeile gesendet, die durch den Wert von ausdruck 2 definiert ist.

Wenn Sie ausdruck 2 weglassen, wird das Programm von der Zeile, die durch den Wert von ausdruck 1 definiert ist, bis zur letzten Zeile gesendet.

Wenn eine Zeile, die dem Wert von ausdruck, ausdruck 1 oder ausdruck 2 entspricht, nicht existiert, so wird jeweils die nächsthöhere Zeile genommen.

Wenn die Zeilen von ausdruck 1 und ausdruck 2 die gleichen sind, erhalten Sie eine Fehlermeldung (ERROR 1).

Das LLIST-Kommando wird ignoriert, wenn ein Passwort gesetzt wurde.

Wurden mit dem MERGE-Kommando mehrere Programme gemischt, so funktioniert das LLIST-Kommando nur für das letzte geladene Programm.

Wenn Sie die vorher gespeicherten Programme auflisten lassen wollen, benutzen

Sie LLIST "etikett",

Die Anzahl der Druckspalten pro Zeile wird mit dem CONSOLE-Kommando festgelegt. Wurde diese Zahl auf 23 oder weniger festgesetzt, erhalten Sie bei der Ausführung des LLIST-Kommandos eine Fehlermeldung (ERROR 3).

LN C,V,F

FORMAT: 1. LN(X)

Abkürzung:
Vergleiche: EXP, LOG

Wirkung:
Errechnet den natürlichen Logarithmus (zur Basis 1Ø) des Ausdrucks X.

Anmerkungen:
Der Wert von X muß größer als Null sein. Wenn Sie den Antilogarithmus einer Zahl finden wollen, benutzen Sie die EXP-Funktion.

Beispiel:

```
1Ø: CLS
2Ø: INPUT "TYPE IN A NUMBER ",X
3Ø: PRINT "THE NATURAL LOGARITHM OF ";X;" IS "; LN (X)
4Ø: INPUT "USE AGAIN? Y/N ",A$
5Ø: IF A$="Y" THEN 2Ø
6Ø: END
```

LOAD C,V,S

FORMAT: 1. LOAD

Abkürzung: LOA.
Vergleiche: OPEN, CLOAD, SAVE

Wirkung:
Lädt Daten über das serielle E/A-Interface (Terminal) in den Programm-/Datenspeicherbereich.

Anmerkungen:
Das LOAD-Kommando ist nur gültig, wenn die Leitung zum seriellen E/A-Interface vorher mit dem OPEN-Kommando geöffnet wurde. Ansonsten wird es ignoriert.

Die Daten werden so lange über das serielle E/A-Interface gelesen, bis der Endcode erreicht ist. Jeder Datensatz bis zum Endcode wird als einzelne Programmzeile interpretiert.

Der PC-1360 wandelt die Daten in eine Form um, die als Programm gespeichert werden kann, und dann liest (schreibt) er ihn in den Programm /Datenspeicherbereich ein. Diese Prozedur wird für jede Programmzeile wiederholt, bis der Textendcode des Textes erreicht wird (siehe OPEN-Kommando).

In einem Datensatz können bis zu 256 Byte gelesen werden. Wenn diese Zahl überschritten wird, erhalten Sie eine Fehlermeldung. Sie erhalten auch Fehlermeldungen, wenn eine einzelne Programmzeile mehr als 80 Byte hat oder nicht mit einem numerischen Wert für die Zeilennummer beginnt.

Zeilen werden nicht nach der Zeilennummer reorganisiert. Die Ausführung des LOAD-Kommandos endet, wenn der Textendcode erreicht wird (vom sendenden Gerät).

Auch wenn das sendende Gerät das ganze Programm sendet, fährt der PC-1360 mit der Ausführung fort solange der Textendcode nicht gelesen wird. In diesem Fall, müssen Sie die Ausführung wie folgt beenden:

(1) Nachdem das sendende Gerät das Programm gesendet hat, muß es auch den Textendcode senden.
(2) Oder drücken Sie die BRK-Taste.

Die Reservespeicherinhalte können nicht über das serielle E/A-Interface gelesen werden.

LOG

C,V,F

FORMAT: 1. LOG(X)

Abkürzung: LO.
Vergleiche: LN

Wirkung:
Errechnet den gemeinen Logarithmus (zur Basis 10) des Ausdrucks X.

Anmerkungen:
Wenn Sie einen Logarithmus zu einer anderen Basis als 10 erhalten wollen, benutzen Sie die folgende Umrechnungsformel: Log zur Basis B von X = LOG(X)/LOG(B).

Wenn Sie den Antilogarithmus eines gemeinen Logarithmus erhalten wollen, erheben Sie 1Ø in die Potenz des Logarithmus mit Hilfe des Operators "^".

Beispiel:

```
LOG(2)
    3.Ø1Ø299957E-Ø1
```

LPRINT AUSGABE ÜBER DAS DRUCKERINTERFACE: C,V,P

LPRINT wird benutzt, um Programmzeilen direkt an den Drucker über das Druckerinterface oder über das serielle E/A-Interface auszugeben.

FORMAT:
1. LPRINT druckausdr
2. LPRINT druckausdr, druckausdr, ...druckausdr
3. LPRINT druckliste
4. LPRINT druckliste;
5. LPRINT

Wo: druckliste ist druckausdr
oder druckausdr; druckliste
Und: druckausdr ist ausdruck
oder USING format; ausdruck

USING wird gesondert unter USING behandelt.

Abkürzung: LP., LPR., LPRI., LPRIN.
Vergleiche: PAUSE, PRINT, USING und WAIT

WIRKUNG:
LPRINT dient dem Ausdrucke von Daten über einen angeschlossenen Drucker (CE-126P oder CE-140P).

ANMERKUNGEN:
Standardmäßig erfolgt die Ausgabe über den angeschlossenen Drucker. Wenn aber der serielle E/A-Port mit dem OPEN-Befehl geöffnet wurde, erfolgt die Ausgabe statt dessen über das serielle E/A-Interface. Der CLOSE-Befehl veranlasst wieder die Ausgabe über den angeschlossenen Drucker.
Beachten Sie, daß der CE-140P zwar an das serielle E/A-Interface angeschlossen ist, er aber wie an das Drucker-Interface angeschlossen arbeitet (der Ausgangskanal sollte also geschlossen sein). Wenn der CE-126P und der CE-140P gleichzeitig angeschlossen sind, erfolgt das Drucken über den CE-126P, während Grafiken über den CE-140P ausgegeben werden.

Die erste Form der LPRINT-Anweisung druckt einen einzelnen Wert aus. Wenn der Ausdruck numerisch ist, wird der Wert ganz rechts am Papierrand ausgedruckt. Wenn es sich um einen Zeichenfolgen-Ausdruck handelt, beginnt der Druck ganz links.

Im Format (2) beschränkt sich die Zahl der Druckspalten auf Gruppen mit je 12 Spalten. Die eingegebenen Werte werden der Reihe nach ausgedruckt. Mit anderen Worten, der erste eingegebene Wert wird links auf der ersten Zeile, der zweite eingegebene Wert wird rechts auf der ersten Zeile, der dritte eingegebene Wert wird links auf der zweiten Zeile und der vierte eingegebene Wert wird rechts auf der zweiten Zeile ausgedruckt.

Die numerischen Werte werden rechtsbündig und der Text linksbündig in den Spalten ausgedruckt. Ist die Ausgabelänge größer als die verfügbare Spaltenbreite, werden die niedrigstwertigen Stellen für die numerische Ausgabe abgerundet und die nachfolgenden Zeichen werden für die Textausgabe abgeschnitten.

Die Anzahl der in Format (2) angegebenen Werte (Posten) muß zwischen 2 und 8 liegen.

Auch wenn die Startposition der Anzeige mit dem Format (4) oder dem CURSOR-Kommando in Format (2) angegeben wurde, wird die Spezifikation gelöscht und der Druck erfolgt in der oben gezeigten Form. Die Werte werden vom linken Papierrand aus ausgedruckt.

In Format (3) werden die Werte links beginnend ausgedruckt. Falls der auszudruckende Wert mehr als 24 Spalten hat, so wird der Druck in der nächsten Zeile fortgesetzt. Maximal können 96 Zeichen ausgedruckt werden. Liegt die 96. Spalte in der Mitte eines numerischen Wertes, erhalten Sie eine Fehlermeldung.

Im Format (4), bei einem angegebenen Druckwert, wird der durch das LPRINT-Kommando angegebene Wert als nächster ausgedruckt. Wegen der Struktur des Druckers jedoch wird der Druck in den folgenden Fällen ausgeführt:

(1) wenn der auszudruckende Wert 24 Spalten überschreitet.
(2) wenn ein LPRINT-Kommando, das nicht mit einem ";" endet, ausgeführt wurde.

(3) wenn die Programmausführung beendet ist.

Im Format (5) wird nichts ausgedruckt, diese Form führt nur zu einem Zeilenvorschub.

Beispiele:

```
1Ø: A = 1Ø:B = 2Ø:X$ = "ABCDE":Y$ = "XYZ"
2Ø: LPRINT A
3Ø: LPRINT X$
4Ø: LPRINT A,B,X$,Y$
5Ø: LPTINT X$;A;B
6Ø: LPRINT
7Ø: LPRINT A*B;
8Ø: LPRINT Y$
```

LPRINT AUSGABE ÜBER DAS SERIELLE E/A-INTERFACE: C,V,S

FORMAT:

1. LPRINT { ausdruck / zeichenfolge }

2. LPRINT { ausdruck / zeichenfolge } , { ausdruck / zeichenfolge } ...,
   { ausdruck / zeichenfolge }

3. LPRINT { ausdruck / zeichenfolge } ; { ausdruck / zeichenfolge } ...;
   { ausdruck / zeichenfolge }

4. LPRINT { ausdruck / zeichenfolge } ;
   (Format, in dem ";" am Ende von 1 und 3 hier oben angehängt wird)

5. LPRINT

Abkürzung: LP.
Vergleiche: OPEN, CONSOLE, USING

Wirkung:
Sendet die angegebene Information über das serielle E/A-Interface (Terminal) aus.

Anmerkungen:
Wenn Sie mit dem OPEN-Kommando die Leitung zum seriellen E/A-Interface geöffnet haben, wird die angegebene Information über das serielle E/A-terminal im ASCII-Code ausgegeben.

Wenn die Leitung geschlossen ist, druckt LPRINT Informationen über den CE-126P oder CE-140P Drucker aus.
Obwohl der CE-140P Drucker an das serielle E/A-Interface angeschlossen ist, arbeitet er wie an das Drucker-Interface angeschlossen.

Im Format (1) wird der Wert von ausdruck oder zeichenfolge von Anfang an gesendet.

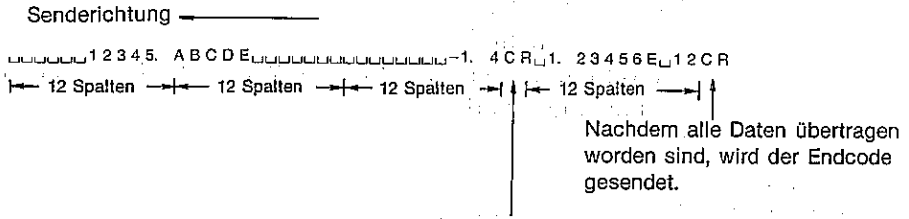
Ist der Wert von ausdruck negativ, so wird vor dem Wert ein Minuszeichen "−" gesendet. Ist er positiv, wird ein Leerzeichen gesendet.

Im Format (2) werden automatisch 12 Spalten(Ziffer)-Unterteiltungen gesetzt. Der Wert eines einzelnen Ausdrucks oder einer Zeichenfolge wird innerhalb eines Bereichs von 12 Spalten (Ziffern) gesendet.

Beispiel:

```
10: OPEN "1200, N, 8, 1, A, C"
20: CONSOLE 36
30: LPRINT 12345, "ABCDE",
    −7/5, 1. 23456789E12
```

Die Ausführung dieses Programms sendet die Daten in folgender Form.


Senderichtung
␣␣␣␣␣12345. ABCDE␣␣␣␣␣

Beispiel:

5Ø: PRINT – 123; "ABC";567.89

←Senderichtung

–123.ABC567.89CR

LF oder CR + LF wird gesendet,
je nachdem wie das OPEN-
Kommando angegeben wird)

Im Format (4) wird der Endcode, der das Ende der Daten angibt, nicht gesendet.

Statt dessen wird der Endcode gesendet, nachdem die Daten, die der durch das CONSOLE-Kommando definierten Anzahl von Spalten entsprechen, gesendet worden sind.

Im Format (5) wird nur der Endcode gesendet.

Wenn das Format mit Hilfe des USING-Kommandos angegeben wurde, senden die Formate (1)—(4) demensprechend.

Die Ausführung von PRINT = LPRINT läßt das PRINT-Kommando als LPRINT-Kommando funktionieren. Die Angabe von PRINT = LPRINT ist nur gültig, wenn der Drucker an das Druckerterminal angeschlossen ist oder wenn die Leitung zum seriellen E/A-Interface mit Hilfe des OPEN-Kommandos geöffnet worden ist.

Um Zeichen oder Kontrollcodes zu senden, nicht direkt über die Tastatur eingegeben werden können, müssen Sie das CHR$-Kommando wie unten angegeben benutzen.

Um [ ] zu senden

(1) 5Ø LPRINT CHR$ &5B, CHR$ &5D

(2) 5Ø A$= CHR$ &5B : B$=CHR$ &5D
    6Ø LPRINT A$, B$

NULL (&ØØ) ist nur in (1) gültig und wird in (2) ignoriert.

LTEXT C,V,Gp

FORMAT: 1. LTEXT

Abkürzung: LT.
Vergleiche: GRAPH

Wirkung:
TEXT wird benutzt, um den den Drucker in den Text-Modus zu setzen.

Anmerkungen:
Dieser Befehl wird benutzt, um den Drucker in den Text-Modus zu setzen, nach der Ausführung des LLIST-Befehls oder nach dem Drucken im manuellen Betrieb mit angeschlossenem CE-140P. Achten Sie darauf, daß Sie den Betriebsmodus entweder mit dem GRAPH-Befehl oder dem LTEXT-Befehl einstellen, wenn die BRK-Taste nach dem automatischen Abschalten des Computers gedrückt wird oder wenn der Hauptschalter des Computers oder des Druckers wieder eingeschaltet wird.

Beispiel:
LTEXT

Setzt den Drucker in den Text-Modus.

MDF C,V,F

FORMAT: 1. MDF(ausdruck)
2. MDF(ausdruck)

Abkürzung: MD.
Vergleiche: USING

Wirkung:
Der MDF-Befehl wird benutzt, um den Wert eines Ausdrucks aufzurunden.

Anmerkungen:
MDF ist eine Funktion, die benutzt wird, um den Wert eines Ausdrucks auf die durch das USING-Kommando angegebene Anzahl von Dezimalstellen aufzurunden.

Dieser Befehl ist nur effektiv, wenn die Anzahl der Dezimalstellen mit Hilfe des USING-Kommandos für einen Wert definiert wurde.

Wenn USING angegeben wurde, werden die Rechnungen mit einer 11stelligen Mantisse ausgeführt. Format 1 benutzt die normale Vorgabeschwellennummer 4. Dies bedeutet, daß wenn die erste Stelle des abgerundeten Teils größer ist als 4, eine Stelle zum nicht abgerundeten Teil übertragen wird. Diese Schwellennummer kann im Format 2 definiert werden.

Beispiel:

	Anzeige
USING"# # #.# # #" MDF (Ø.5/9)	Ø.Ø56
USING"# # #.# # #" MDF (Ø.5/9,5)	Ø.Ø55

```
1Ø: USING"# # #.# # #"
2Ø: A= MDF (5/9)
3Ø: PRINT A
4Ø: USING
5Ø: PRINT A, 5/9
6Ø: END
```

RUN **ENTER**	Ø.556
ENTER	Ø.556 5.55555E-Ø1

MEM

C,V,F

FORMAT: 1. MEM

Abkürzung: M.
Vergleiche: MEM$, SET MEM

Wirkung:
Gibt die Zahl der freien Byte im Programm- und Datenspeicherbereich an.

Anmerkungen:
MEM gibt den freien Programmierbereich im Speicher an. Dieser Wert hängt von dem Gebrauch der RAM-Karten in den Einbauplätzen 1 und 2 ab (siehe MEM$ und SET MEM).

Bitte beachten Sie, daß wenn Sie "D" mit Hilfe von SET MEM einstellen, der ganze Bereich, der für beide RAM-Karten angegeben ist, nicht vollständig frei ist für das Programm und die Daten, und zwar weil die Programm- und Variablenspeicherbereiche auf den respektiven RAM-Karten begrenzt sind. Wenn irgendein Speicher voll besetzt ist, kann der restliche freie Bereich in der anderen Karte nicht benutzt werden.

MEM$ C,V,F

FORMAT: 1. MEM$

Abkürzung: M.$
Vergleiche: MEM, SET MEM

Wirkung:
Gibt an, welche Einbauplätze der RAM-Karten momentan benutzt werden.

Anmerkungen:
MEM$ wird ohne Parameter eingegeben und gibt den Wert "B", "C" oder "D" an. "C" bedeutet, daß nur die RAM-Karte im Einbauplatz 1 benutzt wird. "B" und "D" bedeuten, daß beide RAM-Karten, und somit beide Einbauplätze 1 und 2, benutzt werden. "B" und "D" geben verschiedene Möglichkeiten an, wie die beiden Einbauplätze zusammen verwendet werden. (Siehe SET MEM)

MERGE C,T

FORMAT: 1. MERGE
2. MERGE "dateiname"
(effektivim PROgramm- oder RUN-Modus)

Abkürzung: MER.
Vergleiche: CLOAD

Wirkung:
Mit dem MERGE-Kommando kann ein auf Kassette gespeichertes Programm geladen und mit einem bereits im Speicher vorhandenen Programm gemischt werden.

Anmerkungen:
Mit dem MERGE-Kommando wird das bereits im Speicher des PC-1360 geladene Programm aufbewahrt und ein weiteres Programm von der Kassette in den Rechner geladen. Auf diese Weise können verschiedene Programme gleichzeitig in den PC-1360 geladen werden. Gemischte Programme haben die gleichen Zeilennummern.

Wenn 2 oder mehr Programme im Speicher enthalten sind, beziehen sich die RUN- und GOTO-Kommandos immer auf das letzte zu mischende Programm, außer wenn Sie Programmetiketten verwenden. Dabei wird ein vorher gemischtes Programm mit Hilfe des RUN- oder GOTO-Kommandos, oder der DEF-Taste, zusammen mit dem Etikett ausgeführt. Bitte beachten Sie, daß Sie vorher gemischte Programme nicht aufbereiten können und daß Sie deshalb vor dem Mischen der Programme diesen Etiketten zuweisen müssen. Sind Programmetiketten identisch, so wird das zuletzt gemischte Programm ausgeführt.

Verknüpfung Passwort-geschützter Programme

Wenn Sie Programme mit Passwörtern (Passwort-geschützt) mit dem MERGE-Kommando laden wollen, so müssen wir zwei Fälle unterscheiden je nachdem ob die Programme innerhalb des Computer geschützt sind.

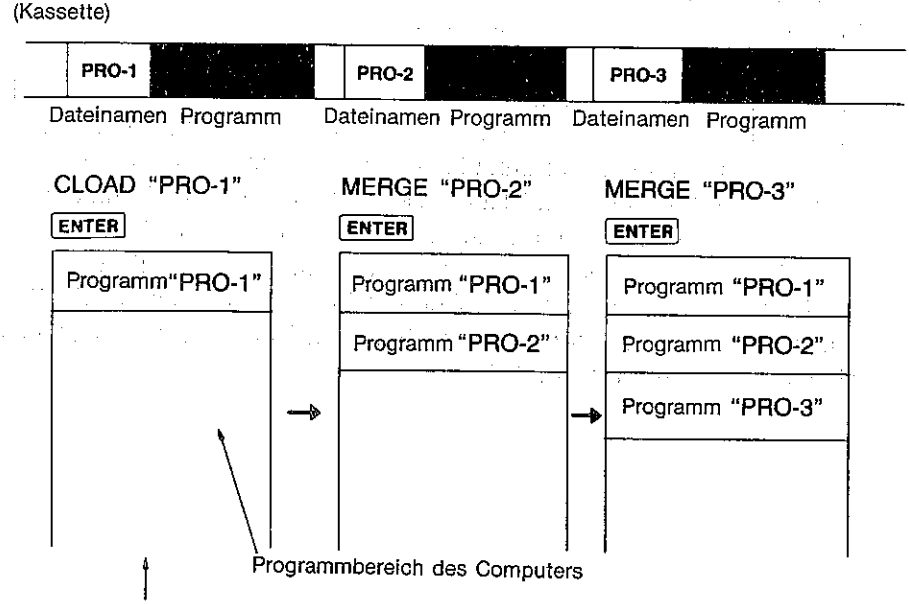
Wenn sie geschützt sind
Passwort-geschützte Programme können nicht geladen werden.

Wenn sie nicht geschützt sind
Wenn Passwort-geschützte Programme mit dem MERGE-Kommando geladen werden, werden alle im Computer enthaltenen Programme zu geschützten Programmen.

Sind die im Computer enthaltenen Programme geschützt, so werden sogar Programme ohne Passwörter zu Passwort-geschützten Programmen, wenn sie mit dem MERGE-Kommando geladen werden.

Beispiel:

Wenn Programme mit den Dateinamen PRO-1, PRO-2 und PRO-3 gespeichert werden sollen, wird das Programm PRO-1 mit dem CLOAD-Kommando eingeladen, und die Programme PRO-2 und PRO-3 werden mit Hilfe des MERGE-Kommandos in den Computer übertragen. Die Abbildung verdeutlicht, wie die Speicherung aus-geführt wird.



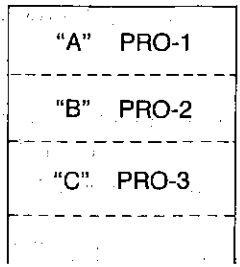
Benutzen Sie das CLOAD-Kommando, um das erste Programm in den Computer zu übertragen.

Programme, die mit dem MERGE-Kommando geladen wurden, werden wie im o.g. Beispiel gespeichert. Ist die erste Zeilennummer des mit dem MERGE-Kommando geladenen Programms größer als die letzte Zeilennummer des vorher geladenen Programms, werden die beiden Programme im folgenden wie eins behandelt.

Ist die erste Zeilennummer des mit dem MERGE-Kommando geladenen Programms kleiner als die letzte Zeilennummer des vorher geladenen Programms, werden die beiden Programme als zwei verschiedene behandelt.

In dem obigen Beispiel, wo die Zeilennummern für die Programme PRO-1, PRO-2 und PRO-3 1Ø-2ØØ, 5Ø-15Ø bzw. 16Ø-3ØØ sind, werden die Programme PRO-1 und PRO-2 als zwei verschiedene Programme behandelt. PRO-2 und PRO-3 werden bei den Zeilennummern 5Ø-3ØØ als ein einziges Programm behandelt.

Ausführen eines gemischten Programmes.



Die Abbildung zeigt die Speicherorganisation, "B" PRO-2 wenn PRO-1 geladen wird, nachdem PRO-2 und "C" PRO-3 mit dem MERGE-Kommando geladen wurden. Wenn nun ein Programm mit RUN oder GOTO (RUN ausdruck oder GOTO ausdruck) gestartet wird, wird das Programm PRO-3 ausgeführt.

Wenn es andererseits mit RUN "etikett", GOTO "etikett" oder einer entsprechenden Taste gestartet wird, wird der Rechner nach dem angegebenen Etikett vom Anfang des Programms PRO-3 suchen. Wenn er es in PRO-3 nicht findet, sucht er weiter im nächsten Programm, das zuletzt gemischt wurde, also in PRO-2.

MID$ C,V,F

FORMAT: 1. MID$(X$,N,M)
2. MID$("zeichenfolge",N,M)

Abkürzung: MI.
Vergleiche: LEFT$, RIGHT$

Wirkung:
Erstellt eine Zeichenfolge von M Zeichen in der Zeichenfolge X$, beginnend beim N-ten Zeichen in der Zeichenfolge X$.

Anmerkungen:
Ist N kleiner als 1 oder größer als die Anzahl von Zeichen in X$, wird eine Null-Zeichenfolge erstellt. M muß zwischen Ø und 8Ø liegen und N muß zwischen 1 und 8Ø liegen. Bruchteile werden abgerundet (abgeschnitten).

Beispiel:
```
1Ø: Z$ = "ABCDEFG"
2Ø: LET Y$ = MID$ (Z$,3,4)
3Ø: PRINT Y$

RUN
CDEF
```

NEXT V

FORMAT: 1. NEXT numerische variable

Abkürzung: N.
Vergleiche: FOR

Wirkung:
NEXT wird benutzt, um das Ende einer Gruppe von Anweisungen zu kennzeichnen, die in einer FOR/NEXT-Schleife wiederholt werden sollen.

Anmerkungen:
Der Gebrauch NEXT ist unter FOR erklärt. Die numerische Variable in einer NEXT-Anweisung muß mit der numerischen Variablen des dazugehörenden FOR übereinstimmen.

Beispiele:
1Ø: FOR I = 1 TO 1Ø
2Ø: PRINT I
3Ø: NEXT I

Druckt die Zahlen von bis 1Ø aus, jedesmal, wenn ENTER betätigt wird.

NEW C

FORMAT: 1. NEW

Abkürzung:
Vergleiche:

Wirkung:
NEW wird benutzt, um bestehende Programme oder den Reservespeicher zu löschen.

Anmerkungen:
Wird es im PROgramm-Modus benutzt, löscht das NEW-Kommando alle Programme und Daten, die sich momentan im Speicher befinden. (Die mit Passwort geschützten Programme können nicht gelöscht werden.)

Wird es im Reserve-Modus benutzt, löscht das NEW-Kommando den bestehenden Reservespeicher.

Das NEW-Kommando ist im RUN-Modus nicht definiert und hat die Fehlermeldung ERROR 9 zur Folge.

Beispiel:
NEW
Löscht das Programm oder den Reservespeicher.

ON...GOSUB V

FORMAT: 1. ON ausdruck GOSUB ausdruck liste
Wo: ausdruck liste ist ausdruck
oder ausdruck, ausdruck liste

Abkürzung: O., GOS.
Vergleiche: GOSUB, GOTO, ON...GOTO

Wirkung:
ON...GOSUB wird benutzt, um eine der gegebenen Subroutinen abhängig vom Wert eines Kontrollausdrucks auszuführen.

Anmerkungen:
Bei der Ausführung von ON...GOSUB wird der ausdruck zwischen ON und GOSUB ausgewertet und auf einen ganzzahligen Wert reduziert. Ist der ganzzahlige Wert 1, so wird die erste Subroutine in der Liste wie in einem normalen GOSUB ausgeführt. Ist der Wert 2, so wird die zweite ausgeführt, und so weiter. Nach dem RETURN aus der Subroutine wird die Anweisung, die direkt auf den ON...GOSUB-Befehl folgt, ausgeführt.

Wenn der Wert des Ausdrucks Null, negativ oder größer als die Zahl der in der Liste gegebenen Subroutinen ist, wird keine Subroutine ausgeführt und und die Programmausführung fährt weiter mit der nächsten Zeile.

In den Ausdrücken nach dem GOSUB dürfen keine Kommata vorkommen. Der PC-1360 kann nicht zwischen Kommata in Ausdrücken und Kommata zwischen Ausdrücken unterscheiden.

Beispiele:

```
 1Ø:  INPUT A
 2Ø:  ON A GOSUB 1ØØ, 2ØØ, 3ØØ
 3Ø:  END
1ØØ:  PRINT "ERSTE"
11Ø:  RETURN
2ØØ:  PRINT "ZWEITE"
21Ø:  RETURN
3ØØ:  PRINT "DRITTE"
31Ø:  RETURN
```

Die Eingabe einer 1 druckt "ERSTE" aus; eine 2 druckt "ZWEITE" aus; eine 3 druckt "DRITTE" aus. Auf andere Eingaben wird nichts ausgedruckt.

ON...GOTO V

FORMAT: 1. ON ausdruck GOTO ausdruck liste
Wo: ausdruck liste ist ausdruck
oder ausdruck, ausdruck liste

Abkürzung: O.: G
Vergleiche: GOSUB, GOTO, ON...GOSUB

Wirkung:
ON...GOTO wird benutzt, um das Programm abhängig von einem Wert an einer bestimmten Stelle fortzusetzen.

Anmerkungen:
Bei der Ausführung von ON...GOTO wird der ausdruck zwischen ON und GOTO ausgewertet und auf einen ganzzahligen Wert reduziert. Ist der Wert dieser Zahl 1, so wird das Programm an der ersten Stelle in der Liste fortgesetzt. Ist der Wert dieser Zahl 2, wird das Programm an der zweiten Stelle fortgesetzt, usw.

Ist der Wert des Ausdrucks Null, negativ oder größer als die Zahl der gegebenen Stellen in der Liste, so wird das Programm auf der nächsten Zeile fortgesetzt.

In den Ausdrücken nach dem GOTO dürfen keine Kommata vorkommen. Der PC-1360 kann nicht zwischen Kommata in Ausdrücken und Kommata zwischen Ausdrücken unterscheiden.

Beispiele:
```
 10: INPUT A
 20: ON A GOTO 100, 200, 300
 30: GOTO 900
100: PRINT "ERSTE"
110: GOTO 900
200: PRINT "ZWEITE"
210: GOTO 900
300: PRINT "DRITTE"
```

```
31Ø: GOTO 9ØØ
9ØØ: END
```

Die Eingabe einer 1 druckt "ERSTE" aus; eine 2 druckt "ZWEITE" aus; eine 3 druckt "DRIT TE" aus. Auf andere Eingaben wird nichts ausgedruckt.

OPEN C,V,S

FORMAT: 1. OPEN "baud rate, parität, wortlänge, stopp bit, code typ, endcode, text end code"
2. OPEN

Abkürzung: OP.
Vergleiche: CLOSE

Wirkung:
Erlaubt die Datenübertragung über das E/A-Interface. Setzt auch die E/A-Optionen.

Anmerkungen:
Format (1) ermöglicht die Datenübertragung über das serielle E/A-Interface (serielle E/A-Terminal). Es setzt auch die Optionen für die Datenübertragung mit der angeschlossenen Ausrüstung. Die Optionen sind in der folgenden Form angegeben: "baud-rate, parität, wortlänge, stoppbit, codetyp, endcode, textendcode"

Baud-Rate:	3ØØ, 6ØØ, 12ØØ Definiert die Schrittgeschwindigkeit (Transfergeschwindigkeit). Für den PC-1360, können Sie 3ØØ Baud, 6ØØ Baud oder 12ØØ Baud auswählen. (1 Baud = 1 bit/sec)
Parität:	N, E, O Definiert den Paritättyp von einem Zeichen. N: Es wird keine Parität gesendet oder empfangen. E: Gibt eine gerade Parität an. O: Gibt eine ungerade Parität an.
Wortlänge:	7,8 Definiert, wieviele Bit pro Zeichen gesendet oder empfangen werden. Sie können entweder 7 oder 8 Bit eingeben.

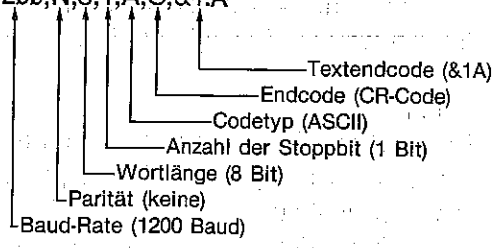
Anzahl der Stoppbit:	1,2
Codetyp:	A Es können nur ASCII-Codes gesendet oder empfangen werden. Deshalb wird immer A angegeben.
Endcode:	C, F, L Definiert den Endcodetyp, um das Ende der Daten (Begrenzung), das Ende der programmzeile usw. anzugeben C: Definiert den CR-Code (Wagenrücklauf) F: Definiert den LF-Code (Zeilenvorschub) L: Definiert den CR-Code + LF-Code.
Textendcode:	&00—&FF Definiert den Textendcode, um das Ende des Programms usw. anzugeben. (Wird beim Gebrauch des SAVE- oder LOAD-Kommandos benötigt.)

Optionen, die im OPEN-Kommando definiert sind, können weggelassen werden. Werden sie weggelassen, bleibt die jetzige Option unverändert.

Im Format (2) werden alle vorher eingestellten Optionen beibehalten. Dieses Format ermöglicht ihnen, Daten über das E/A-Interface zu übertagen.
Wenn Sie das OPEN-Kommando ausführen, während die Leitung zum E/A-Interface geöffnet ist und zur Datenübertagung bereit ist (wegen der vorherigen Ausführung des OPEN-Kommandos), erhalten Sie eine Fehlermeldung (ERROR 8).
Führen Sie das CLOSE-Kommando aus, um die Leitung zu schließen. (Die Leitung wird auch geschlossen, wenn das Programm beendet ist, oder wenn der Computer ausgeschaltet wird.) Die vorher eingestellten Optionen werden beibehalten, auch wenn Sie das CLOSE-Kommando ausführen.

Beispiel:

OPEN "12ØØ,N,8,1,A,C,&1:A"



Die Optionen im obigen Beispiel werden nach dem Auswechseln der Batterien oder nach dem Betätigen des Reset-Schalters eingestellt.

OPEN ",,,2"

Nur die Anzahl der Stoppbit wird geändert

OPEN$ C,V,F,S

FORMAT: 1. OPEN$

Abkürzung: OP.$
Vergleiche: OPEN

Wirkung:
Gibt die momentan gesetzten E/A-Optionen aus.

Anmerkungen:
Die momentan gesetzten E/A-Optionen werden als Zeichenfolge ausgegeben.

Beispiel:
OPEN$ **ENTER** 12ØØ, N, 8, 1, A, C, &1A

PAINT C,V,Gp

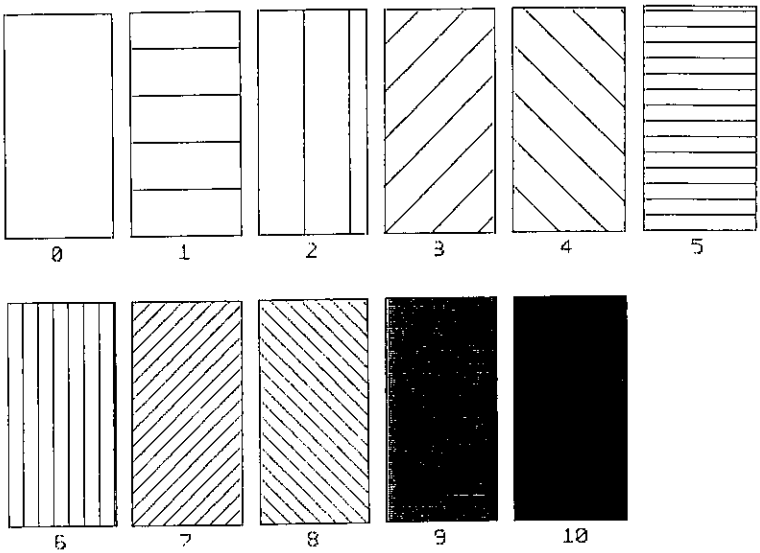
FORMAT: 1. PAINT ausdruck 1 [,ausdruck 2]

Abkürzung: PAI.
Vergleiche: LLINE, RLINE, CIRCLE, COLOR

Wirkung:
PAINT wird benutzt, um das Innere eines Rechtecks oder Kreises (oder Kreissektors) zu schraffieren.

Anmerkungen:
Dieser Befehl ist effektiv nur im Grafik-Modus. Wenn Sie mit einem direkt vor PAINT ausgeführten Befehl ein Rechteck oder ein Kreis (oder Kreissektor) gezeichnet haben, schraffiert der Drucker das Innere des Rechtecks oder Kreises (oder Kreissektors).

Die folgenden Muster können für das Rechteck definiert werden, welches durch die Spezifikation für "B" des LLINE- oder RLINE-Befehls gezeichnet wurde und für den Kreis (oder Kreissektor), der mit dem CIRCLE-Befehl durch Zuweisung eines Wertes für den Ausdruck 1 gezeichnet wurde. Für Rechtecke kann Ø bis 1Ø angegeben werden, während für Kreise (oder Kreissektoren) 1 bis 6, 9 oder 1Ø angegeben werden kann.



Wenn Sie PAINT zusammen mit LLINE wiederholt einsetzen, kann der Drucker ein spezielles Muster zeichnen, wie im Beispiel angegeben.

Ausdruck 2 wird benutzt, um die Farbe der Schraffur anzugeben. Der Wert von ausdruck 2 muß zwischen Ø und 7 im erweiterten Farbcode und zwischen Ø und 3 im normalen Farbcode liegen. (Siehe COLOR für die durch jeden Wert definierte Farbe.)

Wenn Sie PAINT für den mit CIRCLE gezogenen Kreis (oder Kreissektor) ausführen, müssen Sie die Verhältniszahl = 1 und den Teilwinkel = 1 als Werte für ausdruck 7 und ausdruck 8 des CIRCLE-Befehls eingesetzt haben.

Beispiel:

```
LLINE (Ø,Ø) – (2ØØ, –2ØØ), Ø, Ø, B
PAINT 5, Ø
PAINT 6, Ø
```

PASS C

FORMAT: 1. PASS "zeichenfolge"

Abkürzung: PA.
Vergleiche: CSAVE, CLOAD, NEW

Wirkung:
Mit dem PASS-Kommando können Passwörter gesetzt und gelöscht werden.

Anmerkungen:
Passwörter werden benutzt, um Programme vor Einsichtnahme oder Modifikation durch andere Anwender zu schützen. Ein Passwort besteht aus einer Zeichenfolge, die nicht länger als sieben Zeichen enthält. Die sieben Zeichen müssen Buchstaben des Alphabets oder folgende Sonderzeichen sein: ! # $ % & ( ) * + / , . : ; < = > ? @ √ π ∧

Wurde das PASS-Kommando gegeben, so sind die Programme im Speicher geschützt. Ein Passwort-geschütztes Programm kann im Speicher nicht überprüft oder modifiziert werden. Es kann nicht auf Kassette abgespeichert oder mit LIST oder LLIST ausgegeben werden, weiterhin können weder Programmzeilen hinzugefügt noch gelöscht werden. Falls mehrere Programme im Speicher enthalten sind und das PASS-Kommando gegeben wird, sind alle im Speicher enthaltenen Programme geschützt. Die einzige Möglichkeit, diesen Schutz auszuschalten, ist ein weiteres PASS-Kommando mit dem gleichen Passwort einzugeben. Wenn Sie ein Passwort mit mehr als 7 Zeichen eingeben, werden nur die ersten 7 Zeichen als gültig erkannt und zum Setzen und Aufheben des Schutzes benutzt.

Geben Sie **ENTER** gleich nach dem Passwort ein.
Sie erhalten eine Fehlermeldung, wenn Sie nach der Eingabe des Passwortes weitere Zeichen oder Symbole eingeben. In diesem Fall kann das Passwort nicht annulliert werden.
(Beispiel) PASS"ABCDEFG":A = 123 **ENTER** → Fehler

Beispiele:

PASS "SECRET" Setzt für alle Programme im Speicher das Passwort "SECRET"

Passwort beim Gebrauch der RAM-Karte Das Passwort für den PC-1360 und die RAM-Karte sind verschieden.

Wenn Sie die RAM-Karte einstecken, wird ein Passwort zugewiesen, und wenn Sie die RAM-Karte wieder entfernen, wird das Programm auf der RAM-Karte geschützt. Für den PC-1360 wird damit jedoch kein Passwort vergeben.

Darüberhinaus, wenn Sie die RAM-Karte einstecken, nachdem das Programm innerhalb des PC-1360 (d.h. durch Setzen eines Passwortes) geschützt wurde, wird das Programm innerhalb der RAM-Karte nicht geschützt.

(Das zuvor für das Programm auf der RAM-Karte vergebene Passwort bleibt aber weiterhin gültig.)

PAUSE V

FORMAT: 1. PAUSE druck ausdr
2. PAUSE druck ausdr, druck ausdr, ..., druck ausdr
3. PAUSE druck liste
4. PAUSE druck liste;
5. PAUSE

Wo:	druck liste	ist druck ausdr
		oder druck ausdr; druck liste
und:	druck ausdr	ist ausdruck
		oder USING format; ausdruck

Das USING format wird unter USING beschrieben.

Abkürzung: PAU.
Vergleiche: LPRINT, PRINT, CURSOR, USING, WAIT

Wirkung:
PAUSE wird benutzt, um eine Ausgabe auf dem Display zeitlich begrenzt wiederzugegeben.

Anmerkungen:
PAUSE wird benutzt, um kurze Texte, Ergebnisse von Berechnungen usw, anzuzeigen. Die Ausführung von PAUSE ist die gleiche wie die von PRINT, mit der Ausnahme, daß bei PAUSE der PC-1360 eine Pause von ca. .85 Sekunden macht, bevor er mit der Programmausführung fortfährt, ohne auf die ENTER-Taste oder den WAIT-Intervall zu warten.

Die erste Form der PAUSE-Anweisung gibt einen einzelnen Wert aus. Wenn der Ausdruck numerisch ist, wird der Wert ganz rechts ausgedruckt. Bei einem Zeichenfolgen-Ausdruck wird der Wert ganz links beginnend ausgegeben.

Wenn aber die Startposition der Anzeige mit Hilfe des Formats (4) oder des CURSOR-Kommandos angegeben wurde, beginnt die Anzeige an dieser Stelle.

Im Format (2) wird das Anzeigefeld in Gruppen mit je 12 Spalten unterteilt. Die Werte werden der Reihe nach vom ersten spezifizierten Wert an angegeben.
Auch hier wird innerhalb des Blocks von 12 Spalten der numerische Wert eines Ausdrucks ganz rechts und Zeichen ganz links beginnend auf der Anzeige ausgegeben.

Die Anzahl der Werte (Posten) im Format (2) muß zwischen 2 und 8 liegen. Ist der angegebene Wert mehr als 12 Spalten lang, geschieht folgendes:

1) Wenn der Wert mehr als 12 Ziffern lang ist (wenn der Dezimalbruch in der Exponentialanzeige 8 Ziffern oder mehr hat), werden die niedrigstwertigen Stellen abgeschnitten.
2) Wenn die Zeichen 12 Spalten überschreiten, werden nur die ersten 12 Zeichen (von links beginnend) angezeigt.

Im Format (3) wird der angegebene Wert in direkter Folge von der linken Seite des Bildschirms angezeigt. Wurde aber die Startposition der Anzeige mit dem Format (4) oder dem CURSOR-Kommando vorgegeben, beginnt die Anzeige an dieser Stelle.

Wenn der eingegebene Wert in Format (3) das Maximum von 96 Spalten überschreitet, wird der überschüssige Teil nicht angezeigt. Wenn der Wert das Maximum von 96 Spalten überschreitet, wird die Fehlermeldung (ERROR 6) ausgegeben, sofern die 96. Spalte in der Mitte eines numerischen Wertes liegt.

Im Format (4) wird der angegebene Wert von der linken Seite des Bildschirms angezeigt. Die auf das Ende dieses Wertes folgende Zeichen ist die Startposition der nächsten Anzeigebefehle wie z.B. das nächste PRINT-Kommando.

Im Format (5) wird der vorangegangene Wert unverändert wiedergegeben.

Beispiele:

```
1Ø: A = 1Ø:B = 2Ø:X$ = "ABCDEF":
    Y$ = "XYZ"
```

	Anzeige
2Ø: PAUSE A	1Ø.
3Ø: PAUSE X$	ABCDEF
4Ø: PAUSE X$, Y$, A, B	ABCDEF XYZ 1Ø. 2Ø.
5Ø: PAUSE Y$;X$	XYZABCDEF
6Ø: PAUSE A*B	XYZABCDEF2ØØ.

PEEK C,V,M

FORMAT: 1. PEEK speicher adresse

Abkürzung: PE.
Vergleiche: CALL, POKE

Wirkung:
Gibt die Speicherinhalte an der vorgebenene Stelle aus.

Anmerkungen:
PEEK gibt ein Byte des Speichers aus. Der angegebene Wert entspricht 8 Datenbit als Wert zwischen Ø und 255 (&Ø-&FF hexadezimal). Speicheradresse ist der Bereich zwischen Ø bis 65535 (&Ø~&FFFF).

PI C,V,F

FORMAT: 1. PI

Abkürzung:
Vergleiche:

Wirkung:
PI ist eine numerische Pseudovariable, die den Wert PI hat.

Anmerkungen:
Sie ist iden tisch mit dem speziellen PI-Zeichen (π) auf der Tastatur. Gleich wie bei anderen Zahlen wird der Wert von PI mit 1Østelliger Genauigkeit (3,141592654) angegeben.

POINT

C,V,F,Gs

FORMAT: 1. POINT (ausdruck 1, ausdruck 2)

Abkürzung: POI.
Vergleiche: GCURSOR, PSET, PRESET

Wirkung:
Ermittelt den Status des angegebenen Punktes.

Anmerkungen:
Ist der mit (ausdruck 1, ausdruck 2) definierte Punkt gesetzt, so wird eine "1" zurückgegeben, und ist er nicht gesetzt, wird eine "Ø" zurückgegeben. Wenn der definierte Punkt aüerhalb des Bildschirmbereir wird ein "-1" wiedergegeben.

Liegt der definierte Punkt außerhalb der Grenzen des Bildschirmes, wird eine "1" zurückgegeben. Die Werte von ausdruck 1 und ausdruck 2 können zwischen −32768 und +32767 liegen. Die Punkte auf dem Bildschirm aber liegen zwischen Ø und 149 für ausdruck 1 und zwischen Ø und 31 für ausdruck 2.

Beispiel:

```
1Ø: CLS:WAIT Ø:A= 75
2Ø: LINE(5Ø, Ø)−(5Ø, 31)      ┐  2 vertikale Linien werden
                              │  gezogen.
3Ø: LINE (1ØØ, Ø)−(1ØØ, 31)  ┘
4Ø: PSET (A, 16)              ← Setzt einen Punkt zwischen den 2 linien.
5Ø: B= POINT (A+1, 16)        ← Prüft, ob der nächste Punkt rechts gesetzt ist.
6Ø: IF B THEN 15Ø             ← Wenn gesetzt, springe nach 15Ø.
7Ø: PSET (A+1, 16)            ← Wenn nicht gesetzt, setze ihn.
8Ø: PRESET(A, 16)             ← Dann, lösche den alten gesetzten Punkt.
9Ø: A = A + 1                 ← Verschiebe um einen Punkt nach rechts.
```

```
100: GOTO 50          ← Springe zurück nach Zeile 50.
150: B= POINT(A-1, 16) ← Prüft, ob der nächste Punkt links
                         gesetzt ist.
160: IF B THEN 50     ← Wenn gesetzt, springe nach Zeile
                         50.
170: PSET (A-1, 16)   ← Wenn nicht gesetzt, setze ihn.
180: PRESET (A, 16)   ← Dann, lösche den alten gesetzten
                         Punkt.
190: A=A-1            ← Verschiebe um einen Punkt nach
                         links.
200: GOTO 150         ← Springe zurück nach Zeile 150.
```

Dieses Programm bewegt einen Punkt vor und zurück zwischen 2 auf dem Bildschirm angezeigten vertikalen Linien.

POKE C,V,M

FORMAT: 1. POKE speicher adresse, daten [,daten...]

Abkürzung: PO.
Vergleiche: CALL, PEEK

Wirkung:
Schreibt Daten an einer vorgeschriebenen Stelle im Speicher.

Anmerkungen:
POKE schreibt einen oder mehrere Datenbyte, beginnend mit der angegebenen Speicheradresse. Alle Daten müssen als Byte zwischen Ø und 65535 (&Ø-&FFFF hexadezimal) spezifiziert werden. Wenn Sie mehr als ein Byte angeben, werden die folgenden Byte an aufeinanderfolgenden Speicheradressen geschrieben. Wenn Sie die Grenzen der Speicheradresse beim Schreiben überschreiten, erhalten Sie Fehlermeldungen.

PRESET

C,V,Gs

FORMAT: 1. PRESET (ausdruck 1, ausdruck 2)

Abkürzung: PRE.
Vergleiche: PSET, GCURSOR, POINT

Wirkung:
Löscht den definierten Punkt auf dem Bildschirm.

Anmerkungen:
Löscht den durch (ausdruck 1, ausdruck 2) definierten Punkt. Die Werte von ausdruck 1 und ausdruck 2 können im Bereich von −32768 und +32767 liegen. Die Punkte auf dem Bildschirm aber liegen im Bereich von 0 bis 149 für ausdruck 1 und von 0 bis 31 für ausdruck 2.

Beispiel:

```
10: CLS : WAIT 0
20: LINE (20,0)-(130,31),BF
30: FOR X=-25 TO 25 STEP 0.5
40: Y=-1* SQR ABS (25*25-X*X)
50: PRESET (X+75,Y+31)
60: NEXT X
70: WAIT : GPRINT
```

Dieses Programm zeichnet einen Halbkreis in einem gefüllten Quadrat.

PRINT V

FORMAT:
1. PRINT druck ausdr
2. PRINT druck ausdr, druck ausdr, druck ausdr, druck ausdr
3. PRINT druck liste
4. PRINT druck liste;
5. PRINT
6. PRINT = LPRINT
7. PRINT = PRINT

Wo:	druck list	ist	druck ausdr
		oder	druck ausdr; druck liste
und:	druck ausdr	ist	ausdruck
		oder	USING-Format; ausdruck

Das USING-Format wird unter USING beschrieben.

Abkürzung: P.
Vergleiche: LPRINT, PAUSE, CURSOR, USING, WAIT

Wirkung:
PRINT wird zur Ausgabe von Daten Über den Bildschirm oder den Drucker verwendet.

Anmerkungen:
Mit PRINT werden Texte, Ergebnisse usw, ausgegeben. Die erste Form von PRINT gibt einen einzelnen Wert aus. Wenn es ein numerischer Ausdruck ist, wird der Wert auf der rechten Seite des Bildschirms angezeigt. Handelt es sich um einen Zeichenfolgen-Ausdruck, beginnt die Anzeige ganz links.

Wenn aber die Startposition der Anzeige mit Hilfe des Formats (4) oder des CURSOR-Kommandos angegeben wurde, beginnt die Anzeige an dieser Stelle. Im Format (2) wird das Anzeigefeld in Gruppen mit je 12 Spalten unterteilt. Die Werte werden der Reihe nach vom ersten spezifizierten Wert an angegeben. Auch hier werden innerhalb des Blocks von 12 Spalten der numerische Wert eines Ausdrucks ganz rechts und die Zeichen ganz links beginnend auf der Anzeige ausgegeben. Die Anzahl der Werte (Posten) im Format (2) muß zwischen 2 und 8 liegen.

Ist der angegebene Wert mehr als 12 Spalten lang, geschieht folgendes:

1) Wenn der Wert mehr als 12 Ziffern lang ist (wenn der Dezimalbruch in der Exponentialanzeige 8 Ziffern oder mehr hat), werden die niedrigstwertigen Stellen abgeschnitten.
2) Wenn die Zeichen 12 Spalten überschreiten, werden nur die ersten 12 Zeichen (von links beginnend) angezeigt.

Im Format (3) wird der angegebene Wert in direkter Folge von der linken Seite des Bildschirms angezeigt. Wurde aber die Startposition der Anzeige mit dem Format (4) oder dem CURSOR-Kommando vorgegeben, beginnt die Anzeige an dieser Stelle.

Wenn der eingegebene Wert in Format (3) das Maximum von 96 Spalten überschreitet, wird der überschüssige Teil nicht angezeigt. Wenn der Wert das Maximum von 96 Spalten überschreitet, wird die Fehlermeldung (ERROR 6) ausgegeben, sofern die 96. Spalte in der Mitte eines numerischen Wertes liegt.

Im Format (4) wird der angegebene Wert von der linken Seite des Bildschirms angezeigt. Die auf das Ende dieses Wertes folgende Zeichen ist die Startposition der nächsten Anzeigebefehle wie z.B. das nächste PRINT-Kommando.

Kombinieren Sie keinesfalls Bildschirm-Wiedergabekommandos (PRINT usw.) mit seriellen E/A-Kommandos (LPRINT usw. für das serielle E/A-Interface).

Im Format (5) wird der vorangegangene Wert unverändert wiedergegeben.

Die Formen 6 und 7 der PRINT-Anweisung geben nichts aus. Die 6. Form setzt alle PRINT-Anweisungen, die im Programm folgen, gleich LPRINT-Anweisungen. Mit der 7. Form dieser Zu-stand wieder zurückgesetzt, so daß die PRINT-Anweisungen wieder über die Anzeige erfolgen.

Beispiele:

	Anzeige
1Ø: A=123:B=5/9:X$="ABCDEF": Y$="VWXYZ"	**ABCDEF 5.55555E—Ø1**
2Ø: PRINT X$,B	
3Ø: PRINT A;B	**123.5.555555556E—Ø1**
4Ø: PRINT X$;A;	**ABCDEF123.**
5Ø: PRINT Y$;B	**ABCDEF123.VWXYZ5.555555556E—Ø1**

PRINT # C,V,T

FORMAT: 1. PRINT # "var liste"

2. PRINT # "dateiname"; var liste

Wo: var liste ist variable
oder variable, var liste

Abkürzung: P. #
Vergleiche: INPUT #, PRINT, READ

Wirkung:
PRINT # wird benutzt, um Werte auf Kassette abzuspeichern.

Anwendung und Beispiele:
Folgende Variablentypen können für Variablennamen benutzt werden:
(1) Festpunktvariablen—A, B, X, A(26), C✱, A(1Ø)✱, usw.
(2) Einfache Variablen—AA, B2, XY$, usw.
(3) Feldvariablen—B(✱), CD(✱), N$(✱), usw.

1) Speichern des Inhaltes fester Variablen auf Band.
Der Inhalt fester Variablen kann durch Angabe des Namens der entsprechenen Variablen (getrennt durch Kommas) in der PRINT-Anweisung auf Band gespeichert werden.

```
PRINT # "DATA 1";A,B,X,Y
```

Diese Anweisung speichert die Inhalte der Variablen A, B, X und Y in der Banddatei namens "DATA 1" ab.
Wenn Sie den Inhalt der spezifizierten Festpunktvariable und aller darauffolgenden abspeichern wollen, so geben Sie diesen Variablennamen gefolgt von einem Stern (✱) ein.

PRINT # "D-2";D* Diese Anweisung speichert die Inhalte der Variablen D bis Z (und der erweiterten Variablen A(27) und weiter, falls definiert) in der Banddatei namens "D-2" ab.

PRINT # E,X$,A(3Ø)* Diese Anweisung speichert die Inhalte der Festpunktvariablen E und X$ und der erweiterten Variablen A(30) und aller restlichen Variablen auf das Band ohne Dateinamen.

Die indizierten Festpunktvariablennamen A(1) bis A(26) können in der PRINT #-Anweisung in gleicher Weise angegeben werden wie A bis Z (oder A$ bis Z$). Wenn aber das Feld A bereits durch eine DIM-Anweisung definiert wurde, kann A( ) nicht für die Definierung indizierter Festpunktvariablen benutzt werden.

2) Abspeichern der Inhalte von einfachen Variablen (Zwei-Zeichen-Variablen)

Die Inhalte von einfachen Variablen können auf Kassette abgespeichert werden, indem die entsprechenden Variablennamen angegeben werden.

PRINT # "DM-1"; AB, Y1, XY$

Diese Anweisung speichert die Inhalte der einfachen Variablen AB, Y1 und YX$ in der Banddatei namens "DM-1" ab.

3) Abspeichern der Inhalte von Feldvariablen

Die Inhalte aller Variablen eines bestimmten Feldes können auf Kassette abgespeichert werden, indem der mit einem umklammerten Stern (*) indizierte Name des Feldes angegeben wird.

PRINT # "DS-2"; X(*), Y$(*)

Diese Anweisung speichert die Inhalte aller Elemente(X(Ø),X(1),..)des Feldes X und aller Elemente(Y$(Ø),Y$(1),...)des Feldes Y$ in der Banddatei namens "DS-2" ab.

Es ist nicht möglich, die Inhalte eines oder einiger Elemente eines Feldes gesondert zu speichern. Während Festpunktvariablen oder indizierte Festpunktvariablen zulassen, daß nur ein spezifizierter Teil von ihnen gespeichert wird, ist dies mit einem durch die DIM-Anweisung definierten Feld (wie A) nicht möglich.

Wird die PRINT #-Anweisung ohne Variablenamen eingegeben, erhalten Sie eine Fehlermeldung (ERROR 1).

—ACHTUNG—

Die Plätze für erweiterte Variablen wie A(27) und weiter, für einfache Variablen und/oder Feldvariablen müssen vor Ausführung der PRINT#-Anweisung im Programm/Datenspeicherbereich bereitgestellt werden. Sonst ist die Ausführung der PRINT#-Anweisung für nicht definierte Variablen nicht möglich und Sie erhalten eine Fehlermeldung.

PRINT # 1 C,V,S

FORMAT: PRINT#1 variable, variable, variable...

Abkürzung: P.#1
Vergleiche: OPEN, INPUT#1

Wirkung:
Sendet die Inhalte der vorgegebenen Variablen über das serielle E/A-Interface (Terminal).

Anmerkungen:
Dieses Kommando ist nur effektiv, wenn die Leitung zum seriellen E/A-Interface (mit Hilfe des OPEN-Kommandos) geöffnet wurde. Ansonsten wird es ignoriert.
* Variablen werden folgendermaßen vorgegeben:

Festpunktvariablen: Jeder Variablennamen muß angegeben werden.
[Beispiel] A, B, C$
Anmerkung: Festpunktvariablen können nicht in der Form A* angegeben werden.

Einfache Variablen: Jeder Variablennamen muß angegeben werden.
[Beispiel] AA, B1$, C2

Feldvariablen: Müssen in der Form eines Feldnamens (*) angegeben werden.
[Beispiel] B(*), C$(*)
In dieser Form werden die Inhalte aller Elemente eines Feldes übertragen. (Einzelne Feldelemente können nicht angegeben werden.)
[Beispiel] 5Ø PRINT#1A, AB, C$, E(*).

Bei der Datenübertragung wird der Endcode am Ende der Inhalte jeder Variablen hinzugefügt. Der Endcode wird auch am Ende der Inhalte jedes Elementes für Feldvariablen hinzugefügt.

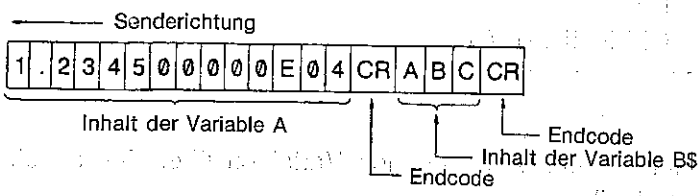
[Beispiel] Wenn

A = 12345 und B$ = "ABC"

und Sie

PRINT#1A,B$

werden A und B$ in der folgenden Form gesendet. (Vorausgesetzt der Endcode is CR.)



* Ist der Wert der numerischen Variablen negativ, so wird vor der Zahl ein Minuszeichen "−" gesendet.

Die Elemente eines Feldes werden in der folgenden Sequenz übertragen.

1-dimensionales Feld

[Beispiel] für B (3)

B (Ø) → B (1) → B (2) → B (3)

2-dimensionales Feld

[Beispiel] für C (2, 3)

C (Ø, Ø) → C (Ø, 1) → C (Ø, 2)...

Die Plätze für erweiterte Variablen wie A(27) und weiter, sowie für einfache Variablen und/oder Feldvariablen müssen im Programm-/Datenspeicherbereich abgelegt sein, bevor das PRINT#1-Kommando ausgeführt wird. Sie erhalten eine Fehlermeldung, wenn Sie versuchen den Inhalt einer Variablen, die nicht zugewiesen worden ist, zu senden.
Sie erhalten auch eine Fehlermeldung, wenn der Typ der vorgegebenen Variable (numerisch oder Zeichenvariable) nicht mit dem im PC-1360 übereinstimmt. Sind in den zu sendenden Daten Sonderzeichen wie "π" und "$\sqrt{\ }$" enthalten, werden diese in "PI" bzw. "SQR" umgewandelt und dann gesendet.

PSET

C,V,Gs

FORMAT: 1. PSET (ausdruck 1, ausdruck 2)
2. PSET (ausdruck 1, ausdruck 2), X

Abkürzung: PS.
Vergleiche: PRESET, GCURSOR, POINT

Wirkung:
Setzt oder invertiert den definierten Punkt auf dem Bildschirm.

Anmerkungen:
Format (1) setzt den durch (ausdruck 1, ausdruck 2) definierten Punkt. Format (2) löscht den durch (ausdruck 1, ausdruck 2) definierten Punkt, wenn dieser bereits gesetzt ist, und setzt ihn, wenn er noch nicht gesetzt ist. Die Werte von ausdruck 1 und ausdruck 2 können im Bereich von −32768 bis +32767 liegen. Die Punkte auf dem Bildschirm aber liegen im Bereich von Ø bis 149 für ausdruck 1 und von Ø bis 31 für ausdruck 2.

Beispiel:

```
1Ø: CLS : WAIT Ø: DEGREE
2Ø: FOR A = Ø TO 6ØØ
3Ø: B = −1 * SIN A
4Ø: Y = INT (B*16) + 16
5Ø: X = INT (A/4)
6Ø: PSET (X, Y)
7Ø: NEXT A
8Ø: WAIT : GPRINT
```

Dieses Programm erzeugt eine Sinuskurve auf dem Bildschirm.

RADIAN C,V

FORMAT: 1. RADIAN

Abkürzung: RAD.
Vergleiche: DEGREE, GRAD

Wirkung:
RADIAN wird benutzt, um die Darstellung der Winkelwerte in die Radialform umzuwandeln.

Anmerkungen:
Der PC-1360 hat drei Möglichkeiten der Darstellung von Winkelwerten - Alt-Grad, Radial-Werte und Neu-Grad. Diese Formen werden für die Argumente der Funktionen SINus, COSinus und TANgens und für die Ergebnisse der Umkehrfunktionen ArcusSiNus, ArcusCoSinus und ArcusTaNgens gebraucht.

Die RADIAN-Funktion wechselt die Darstellungsform aller Winkelwerte auf Radialwert, bis ein DEGREE- oder GRAD-Befehl eingesetzt wird. Die Radialform gibt die Winkel als Bogenmaß in Abhängigkeit vom Radius wieder, z.B. 360° sind 2 PI, da der Einheitskreis den Umfang 2 mal PI mal Radius hat.

Beispiele:
```
1Ø: RADIAN
2Ø: X= ASN 1
3Ø: PRINT X
```

X hat den Wert 1,57Ø796327 oder PI/2, der Arcsinus von 1.

RANDOM C,V

FORMAT: 1. RANDOM

Abkürzung: RA.
Vergleiche: RND

Wirkung:
RANDOM wird benutzt, um eine neue Startzahl für den Zufallszahlengenerator zu setzen.

Anmerkungen:
Wenn mit der RND-Funktion Zufallszahlen erzeugt werden, so startet der PC-1360 bei einer vorgegebenen Startzahl. Der RANDOM-Befehl setzt diese Startzahl auf einen neuen, zufällig gewählten Wert.

Die Startzahl ist jedesmal die gleiche, wenn der PC-1360 eingeschaltet wird, so daß sich die Sequenz der Zufallszahlen mit RND jedesmal wiederholt, es sei denn, die Startzahl wird gewechselt. Diese Eigenschaft ist sehr wichtig bei der Entwicklung eines Programms, denn dies bedeutet, daß das Verhalten des Programms immer gleich sein muß, jedesmal, wenn es abläuft, obwohl es eine RND-Funktion enthält. Wenn Sie wirklich zufällige Zahlen haben wollen, kann mit der RANDOM-Anweisung die Startzahl selbst zufällig ermittelt werden.

Beispiele:
1Ø: RANDOM
2Ø: X = RND 1Ø

Bei Start in Zeile 20 beruht der Wert von X auf einer Standard-Startzahl. Bei Start in Zeile 1Ø wird eine neue Basisermittelt.

READ V

FORMAT: 1. READ variable liste
Wo: variable liste ist variable
oder variable, variable liste

Abkürzung: REA.
Vergleiche: DATA, RESTORE

Wirkung:
READ wird gebraucht, um Werte aus der DATA-Anweisung herauszulesen und sie Variablen zuzuweisen.

Anmerkungen:
Wenn einem Feld Basiswerte zugewiesen werden sollen, so ist es sinnvoll, diese Werte in einer DATA-Anweisung unterzubringen und sie von dort mit einer READ-Anweisung in einer FOR...NEXT-Schleife in das Feld zu übertragen. Wird der erste READ-Befehl ausgeführt, so wird der erste Wert in der ersten DATA-Anweisung zugewiesen. Die folgenden READ-Befehle benutzen die folgenden Werte in der Reihenfolge ihrer Erscheinung im Programm, unabhängig davon, wieviele Werte in jeder DATA-Anweisung angegeben oder wieviele DATA-Anweisungen eingesetzt werden.

Wenn nötig, können dieselben Werte in einer DATA-Anweisung ein zweites Mal mit Hilfe der RESTORE-Anweisung gelesen werden.

Beispiel:
```
10: DIM B (10)
20: WAIT 32
30: FOR I=1 TO 10
40: READ B(I)
50: PRINT B(I)*2;
60: NEXT I
70: DATA 10, 20, 30, 40, 50, 60
80: DATA 70, 80, 90, 100
90: END
```
[10] Dimensionierung eines Feldes
[40] Lädt die Werte aus der DATA-Anweisung in B( ) ein; B(1) ist 10, B(2) ist 20, B(3) ist 30 usw.

REM V

FORMAT: 1. REM kommentar

Abkürzung:
Vergleiche:

Wirkung:
REM wird benutzt, um Kommentare in ein Programm einzufügen.

Anmerkungen:
Öfters ist es sinnvoll, in ein Programm erläuternde Kommentare einzufügen. Es kann sich hierbei um Titel, Autorennamen, Daten der letzten Änderungen, Anwendungshinweise, Erinnerungshinweise für Algorithmen usw. handeln. Solche Kommentare können mit Hilfe der REM-Anweisung eingefügt werden.

Die REM-Anweisung hat keinen Einfluß auf den Programmablauf und kann überall im Programm eingefügt werden. Alles, was in dieser Zeile nach dem REM-Befehl kommt, wird als Kommentar behandelt.

Beispiele:
1Ø: REM DIESE ZEILE HAT KEINEN EFFEKT

RENUM C

FORMAT: 1. RENUM [neue zeile#], [alte zeile#],[inkement]

Abkürzung: REN.
Vergleiche: DELETE, LIST

Wirkung:
Führt die Neunumerierung der Zeilen eines Programms aus.

Anmerkungen:
Die Zeilennummern werden mit dem vorgegebenen inkrement von alte zeile auf neue zeile geändert. Wird neue zeile nicht angegeben, werden die Zeilen neu numeriert, beginnend mit 1Ø in inkrementen von 1Ø. RENUM aktualisiert die Bezugszeilennummern in den Anweisungen GOTO, ON...GOTO, GOSUB, ON...GOSUB, RESTORE und (IF)...THEN.

Ist eine Zeilennummer größer als 65279, oder existiert eine angegebene Zeilennummer nicht, erhalten Sie die Fehlermeldung ERROR 4. Wenn Sie die Reihenfolge der Ausführung ändern, erhalten Sie die Fehlermeldung ERROR 1. RENUM wird nur in den BASIC- und PRO-Modi verwendet (andere Modi: Fehlermeldung ERROR 9.) Wenn Sie ein Passwort benutzt haben, wird das Kommando ignoriert und das Aufforderungszeichen erscheint wieder.

Wenn mehrere Programme mit MERGE im Speicher gemischt wurden, so führt RENUM nur die Neunumerierung des letzten gemischten Programms aus. Falls die niedrigste Zeilennummer des neu numerierten Programms größer ist als die höchste Zeilennummer des zuletzt gemischten Programms, werden die beiden Programme von da an als ein einziges Programm interpretiert.

Enthält eine Zeilennummer nichtnumerische Zeichen, erhalten Sie die Fehlermeldung ERROR 9.
Beispiele von Fehlermeldungen:

```
GOTO 1+2              GOTO ABS (-100)
GOTO"A"+"B"           GOTO LEFT$ ("ABC",2)
```

GOTO A
GOTO 1ØØ.Ø

GOTO B$
GOTO + 1EØ2

Sie erhalten die Fehlermeldung ERROR 3, wenn bei der Neunumerierung die Zeilenlänge größer als 79 Byte ist.

Wenn auf der Anzeige das Zeichen "*" erscheint, drücken Sie die **BRK**-Taste, um die Neunumerierung zu unterbrechen. Wenn "**" erscheint, bedeutet es, daß die Unterbrechung nicht möglich ist. Die Fehlererzeugung oder der Gebrauch der **BRK**-Taste läßt das Programm unverändert.

RENUM-Fehlerliste

Fehleranzeige	Beschreibung	Anzeige durch blinkenden Cursor
ERROR 1	RENUM—Eingabefehler	Fehlerstelle
ERROR 1 IN	":", "," oder CR nach GOTO oder GOSUB; oder ":" oder CR nach THEN	Fehlerstelle; oder Ende der Zeile
ERROR 3 IN	neue Zeilennummer größer als 65279; oder Zeilenlänge größer als 79 Byte	Zeilenkopf; oder Ende der Zeile
ERROR 4	Bezugszeile existiert nicht; oder ein Parameter ist größer als 65279	Fehlerstelle
ERROR 4 IN	Bezugszeile existiert nicht	Bezugszeile
ERROR 9	Versuch RENUM im RUN- oder TEXT-Modus auszuführen	Fehlerstelle
ERROR 9 IN	nichtnumerische Zeichen in der Zeilennummer; oder nichtausführbare Anweisung nach THEN	Fehlerstelle

Beispiel:

```
1Ø: INPUT "CONTINUE";A$
2Ø: IF A$="YES THEN 1Ø
3Ø: IF A$="NO THEN 6Ø
4Ø: PRINT "ENTER YES OR NO PLEASE!"
5Ø: GOTO 1Ø
6Ø: END

RENUM 1ØØ,1Ø,5
LIST

1ØØ: INPUT"CONTINUE";A$
1Ø5: IF A$="YES THEN 1ØØ
11Ø: IF A$="NO THEN 125
115: PRINT"ENTER YES OR NO PLEASE!"
12Ø: GOTO 1ØØ
125: END
>
```

RESTORE V

FORMAT: 1. RESTORE
2. RESTORE ausdruck

Abkürzung: RES.
Vergleiche: DATA, READ

Wirkung:
RESTORE wird benutzt, um Werte in einer DATA-Anweisung mehrmals zu lesen oder die Reihenfolge, in der diese Werte gelesen werden sollen, zu ändern.

Anmerkungen:
Beim normalen Gebrauch von READ beginnt der PC-1360 das Lesen beim ersten Wert in einer DATA-Anweisung und geht sequentiell durch alle übrigen Werte. Die erste Form der RESTORE-Anweisung setzt den Zeiger auf den ersten Wert der ersten DATA-Anweisung zurück, damit er noch einmal gelesen werden kann. Die zweite Form der RESTORE-Anweisung setzt den Zeiger auf den ersten Wert der ersten DATA-Anweisung zurück, deren Zeilennummer größer als der Wert des Ausdrucks ist.

Beispiele:
```
1Ø: DIM B(1Ø)
2Ø: WAIT 32
3Ø: FOR I=1 TO 1Ø
4Ø: RESTORE
5Ø: READ B(I)
6Ø: PRINT B(I)*I;
7Ø: NEXT I
8Ø: DATA 2Ø
9Ø: END
```

[1Ø] Dimensionierung eines Feldes
[5Ø] READ B(I) Weist jedem Element von B( ) den Wert 2Ø zu

RETURN V

FORMAT: 1. RETURN

Abkürzung: RE.
Vergleiche: GOSUB, ON...GOSUB

Wirkung:
RETURN wird am Ende einer Subroutine benutzt, um zur Anweisung nach dem ursprünglichen GOSUB-Befehl zurückzuspringen.

Anmerkungen:
Eine Subroutine kann mehr als eine RETURN-Anweisung enthalten, aber die erste, die ausgeführt wird, beendet die Ausführung der Subroutine. Die nächste Anweisung, die ausgeführt wird, ist dann die Anweisung nach dem GOSUB- oder ON...GOSUB-Befehl, die die Subroutine aufruft. Wird ein RETURN ohne GOSUB ausgeführt, erhalten Sie die Fehlermeldung ERROR 5.

Beispiele:
```
 1Ø: GOSUB 1ØØ
 2Ø: END
1ØØ: PRINT "HELLO"
11Ø: RETURN
```

Das Programm druckt das Wort "HELLO" einmal aus.

RIGHT$ C,V,F

FORMAT: 1. RIGHT$(X$,N)
2. RIGHT$("zeichenfolge",N)

Abkürzung: RI.
Vergleiche: LEFT$, MID$

Wirkung:
Erstellt eine Zeichenfolge von N Zeichen in der Ziechenfolge X$, beginnend beim rechten Ende der Zeichenfolge.

Anmerkungen:
Der Wert N muß zwischen Ø und 8Ø liegen. Bruchteile werden abgerundet (abgeschnitten). Ist N kleiner als 1, wird eine Null-Zeichenfolge herausgenommen. Ist N größer als die Anzahl der Zeichen in X$, wird die ganze Zeichenfolge herausgenommen.

Beispiel:
```
 5: WAIT 32
1Ø: XX$ = "SHARP COMPUTER"
2Ø: FOR N = 1 TO 15
3Ø: LET SS$ = RIGHT$(XX$,N)
4Ø: PRINT SS$
5Ø: NEXT N
```

```
RUN
R
ER
TER
UTER
PUTER
MPUTER
OMPUTER
COMPUTER
 COMPUTER
P COMPUTER
RP COMPUTER
ARP COMPUTER
HARP COMPUTER
SHARP COMPUTER
SHARP COMPUTER
>
```

RLINE

C,V,Gp

FORMAT: 1. RLINE (ausdruck 1, ausdruck 2) — (ausdruck 3, ausdruck 4)[,ausdruck 5][,ausdruck 6][,B]

Abkürzung: RL.
Vergleiche: LLINE, PAINT, COLOR

Wirkung:
RLINE wird benutzt, um eine Linie zwischen zwei durch relative Koordinaten definierte Punkte zu ziehen.

Anmerkungen:
Dieser Befehl ist nur effektiv im Grafik-Modus.

RLINE unterscheidet sich von LLINE darin, daß LLINE den Ursprung der durch SORGN angegebenen Koordinaten als Bezugspunkt nimmt und die Stelle von jedem Punkt durch Koordinaten in Bezug auf diesen Ursprungspunkt angibt, während RLINE die jetzige Position des Zeichenstifts als Ursprung der Koordinaten nimmt und die Stelle des nächsten Punktes durch Koordinaten in Bezug auf diesen Ursprungspunkt angibt.

Beachte:
Wenn nach der Ausführung dieser Befehle mit B LLINE oder RLINE ohne B ausgeführt werden, sollte der nachfolgende Befehl unmittelbar nach der Ausführung dieser Befehle ohne B ausgeführt werden.

```
POKE &FB2Ø,Ø
```

Wenn dieser Befehl nicht ausgeführt wird, mag der Computer ein Rechteck mit LLINE oder RLINE ohne B zeichnen.

Beschreibungen der Spezifikation sind die gleichen wie bei LLINE außer, daß (ausdruck 1, ausdruck 2) nicht weggelassen werden kann.

Beispiel:

```
 5: OPEN
1Ø: GRAPH
2Ø: FOR A=1 TO 3
3Ø: RLINE (Ø,−4Ø) − (6Ø,−5Ø)
4Ø: NEXT A
5Ø: LTEXT
6Ø: LPRINT
7Ø: END
```

RND

C,V,F

FORMAT: 1. RND numerischer ausdruck

Abkürzung: RN.
Vergleiche: RANDOM

Wirkung:
RND ist eine numerische Funktion, die Zufalsszahlen erzeugt.

Anmerkungen:
Ist der Wert des Arguments kleiner als 1, aber größer oder gleich Null, so ist die Zufallszahl kleiner als 1 und größer oder gleich Null. Ist das Argument eine ganze Zahl, die größer oder gleich 1 ist, ist das Ergebnis eine Zufallszahl, die größer oder gleich 1 oder kleiner oder gleich dem Argument ist. Ist das Argument größer oder gleich 1 und keine ganze Zahl, ist das Ergebnis eine Zufallszahl, die größer oder gleich 1 und kleiner oder gleich der kleinsten ganzen Zahl ist, die wiederum größer als das Argument ist: (In diesem Fall ändert sich die Erzeugung der Zufallszahl, je nach dem Wert des Dezimalteils des Arguments.):

	..... Ergebnis	
Argument	Unter Grenze	Obere Grenze
.5	Ø<	<1
2	1	2
2.5	1	3

Normalerweise wird die gleiche Sequenz von Zufallszahlen erzeugt, da jedes Mal die gleiche Basiszahl benutzt wird, wenn der PC-1360 eingeschaltet wird. Um diese Basiszahl zufällig zu ändern, brauchen Sie den RANDOM-Befehl.

Beispiel:
RUN 1ØØ

Führt das Programm aus, beginnend bei Zeile 1ØØ.

RUN C

FORMAT: 1. RUN
2. RUN seilennummer

Abkürzung: R.
Vergleiche: GOTO, MERGE, ARUN

Wirkung:
Mit dem RUN-Kommando wird das Programm im Speicher gestartet.

Anmerkungen:
Die erste Form des RUN-Kommandos startet ein Programm mit der Zeile, die die niedrigste Zeilennummer hat.

Die zweite Form des RUN-Kommandos startet ein Programm mit der angegebenen Zeilennummer.

Wenn mehrere Programme im Speicher enthalten sind, weil Programme mit MERGE gemischt wurden, führt das RUN-Kommando das zuletzt gemischte Programm aus, außer wenn im Programm und im Kommando ein Programmetikett benutzt wird.

GOTO unterscheidet sich in acht Punkten vom RUN-Kommando:
1) Der Wert des WAIT-Intervalles wird zurückgesetzt.
2) Wurde die Anzeige durch USING-Anweisungen formatiert, wird sie gelöscht.
3) Variablen und Felder, mit Ausnahme der festen Variablen, werden gelöscht.
4) Der PRINT = LPRINT-Status wird gesetzt.
5) Der READ-Zeiger wird zurückgesetzt an die erste DATA-Anweisung.
6) Die Cursor-Spezifikationen werden gelöscht.
7) Der grafische Cursor wird auf (Ø, 7) eingestellt.
8) Der Anschluß des seriellen E/A-Interface ist geschlossen.

Die Ausführung eines Programms mit dem Kommando GOTO ist die gleiche wie die Ausführung mit der DEF-Taste. Alle drei Formen des Programmstarts löschen FOR/NEXT- und GOSUB-Gruppen.

SAVE C,V,S

FORMAT: 1. SAVE

Abkürzung: SA.,
Vergleiche: OPEN, LLIST, LOAD, CSAVE

Wirkung:
Sendet das Programm aus dem PC-1360 über das serielle E/A-Interface (Terminal).

Anmerkungen:
Wenn die Leitung zum seriellen E/A-Interface mit dem OPEN-Kommando geöffnet ist, wird das Programm im ASCII-Code übertragen. Das Kommando wird ignoriert, wenn die Leitung geschlossen ist.

Nachdem das ganze Programm übertragen wurde, wird der Textendcode gesendet.

Das SAVE-Kommando wird ignoriert, wenn ein Passwort gesetzt wurde.

SET MEM C

FORMAT: 1. SET MEM "C"
2. SET MEM "B"
3. SET MEM "D"

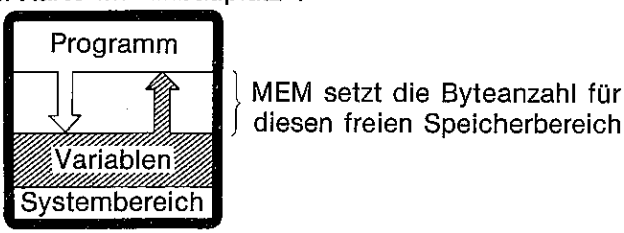
Abkürzung: SE. M.
Vergleiche: MEM, MEM$

Wirkung:
SET MEM setzt fest, welche RAM-Einbauplätze benutzt werden, und setzt die Speicherstruktur für den Programmbereich und den Datenvariablenbereich.

Anmerkungen:
Lesen Sie im Kapitel nach, wo der Gebrauch der RAM-Einbauplätze beschrieben ist.

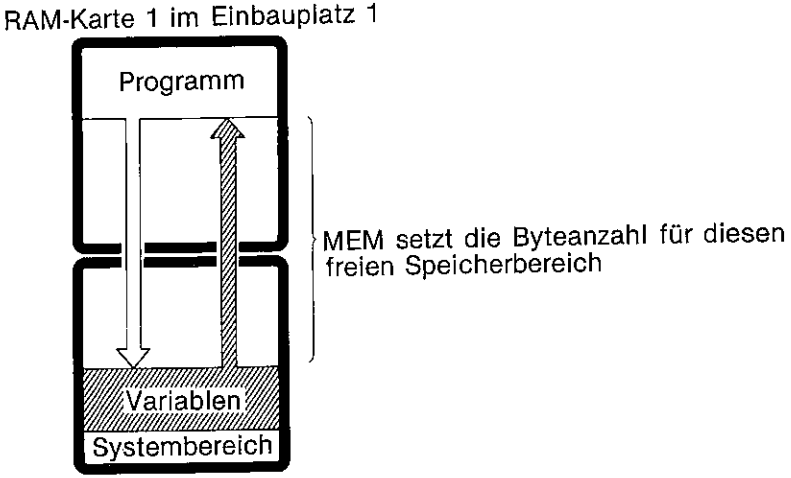
1. SET MEM "C"
Dieses Format benutzt nur den RAM-Einbauplatz 1. Das Programm wird im oberen Teil gespeichert und der Variablenbereich wird unten reserviert. Benutzen Sie diesen Modus, wenn Sie nur eine RAM-Karte verwenden wollen.

RAM-Karte im Einbauplatz 1



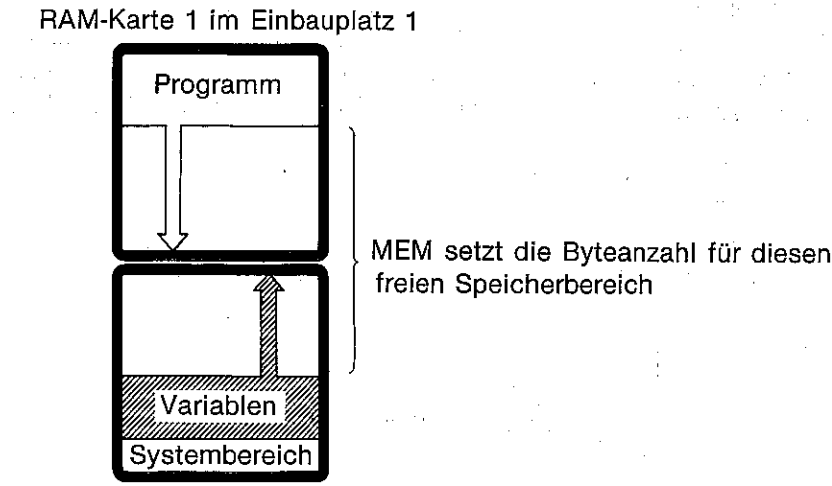
2. SET MEM "B"
Dieses Format erlaubt Ihnen beide RAM-Karten gleichzeitig zu benutzen. Der Programmteil wird an der Startposition in der Karte im Einbauplatz 1 gehalten, während der Systembereich und der Variablenteil an der Startposition in der Karte im Einbauplatz 2 gehalten werden. Sowohl der Programmteil als auch der Variablenteil kann in den anderen Speicher

übergreifen. Diese Anordnung erlaubt den höchstflexiblen Gebrauch des gesamten freien Speichers, bedeutet aber, daß die beiden Karten immer zusammen verwendet werden müssen. Der Gebrauch nur einer Karte allein oder das Auswechseln einer Karte für eine andere ist nicht möglich.



3. SET MEM "D"

Diese Format erlaubt Ihnen auch beide RAM-Karten gleichzeitig zu benutzen. Der Programmteil wird nur in der Karte im Einbauplatz 1 gehalten und der Variablenteil wird nur in der Karte im Einbauplatz 2 gehalten. Der Überlauf von einem Bereich in den Speicherbereich der anderen Karte ist nicht erlaubt. Dies schränkt die Flexibilität für den gemeinsamen Gebrauch beider Karten ein. Falls der benutzte Speicherbereich die Kapazität einer der beiden Karten überschreitet, erhalten Sie eine Fehlermeldung. Der Vorteil dieser Struktur ist, daß Karten einzeln ausgetauscht werden können. Zum Beispiel können Sie eine Karte gegen eine andere austauschen, die mit dem gleichen SET MEM "D"-Format benutzt wurde. Bitte beachten Sie, daß die Karte, die das Programm enthält, immer im Einbauplatz 1 und die Karte, die die Variable enthält, immer im Einbauplatz 2 sein müssen.



Speicherwechsel bei den RAM-Karten:
Diese Operation ist nur um RUN-Modus möglich.

Von "B" oder "D" nach "C"
Mit dieser Operation werden die Programm- und Variablenteile, die in zwei Karten enthalten sind, auf eine einzige Karte übertragen. Wenn der Gesamtbereich den freien Speicherbereich auf einer Karte überschreitet, wird die Ausführung gestoppt und Sie erhalten eine Fehlermeldung.

Von "D" nach "B"
Eliminiert die Beschränkung, daß die Programm- und Variablenbereiche in ihren respektiven Kartenspeichern enthalten sein müssen.

Von "B" oder "C" nach "D"
Sind das Programm oder die Daten zu groß, um auf nur einer einzigen Karte gespeichert zu werden, erhalten Sie eine Fehlermeldung.

Von "C" nach "B"
Verteilt die Programm- und Variablenteile, die auf einer einzigen Karte enthalten sind, auf zwei Karten. Schalten Sie zuerst den Computer aus, legen die neue Karte in den Einbauplatz 2 ein, schalten das Gerät wieder an und führen SET MEM "B" aus.

Bestimmung des gesamten Programmbereichs:
Geben Sie im RUN-Modus den CLEAR-Befehl ein und drücken die ENTER-Taste.
Für die Speicherstruktur "C":
(Kapazität der RAM-Karte 1 (KB) × 1024) − MEM-1634

Für die Speicherstruktur "B" und "D":

((Kapazitäten der RAM-Karte 1 + RAM-Karte 2 (KB) × 1024) − MEM-1634

Wenn sie eine Karte herausnehmen, vergewissern Sie sich zuerst mit eingeschaltetem Gerät, daß die Struktur entweder "C" oder "D" ist (mit Hilfe von MEM$), schalten dann das Gerät aus und nehmen die Karte heraus. Vergessen Sie nicht, daß Karten nicht herausgenommen werden können, wenn MEM$ die Speicherstruktur "B" angibt.

RAM-Karten, die Programme enthalten, weiche mit den Computern PC-1450 und PC-1460 erstellt wurden, können direkt auf dem PC-1360 eingesetzt werden. Sie können auch Karten auf verschiedenen PC-1360 Modellen austauschen und verwenden.

Programme und Datan werden nicht gelöscht, wenn Sie SET MEM benutzen, um von einer Speicherstruktur zur anderen umzuschalten.

Die inhalte der Karte im Einbauplatz 1 können auf die Karte im Einbauplatz 2 kopiert werden (lesen Sie das Kapitel über den Gebrauch der RAM-Karten).

SGN C,V,F

FORMAT: 1. SGN(X)

Abkürzung: SG.
Vergleiche:

Wirkung:
Erstellt das Vorzeichen des Ausdrucks X.

Anmerkungen:
X kann jeder beliebige numerische Ausdruck sein. SGN(X) erstellt entweder 1, 0 oder −1, abhängig vom Vorzeichen des Ausdrucks X wie hier unten angegeben:

Wert von X	Von SGN(X) erstellter Wert
X>Ø	1
X=Ø	Ø
X<Ø	−1

Beispiel:

```
 5: WAIT 1ØØ
1Ø: FOR N=−3 TO 3
2Ø: PRINT N, SGN(N)
3Ø: NEXT N
4Ø: END
```

RUN

```
−3.          −1.
−2.          −1.
−1.          −1.
 Ø.           Ø.
 1.           1.
 2.           1.
 3.           1.
```

SIN

C,V,F

FORMAT: 1. SIN(X)

Abkürzung: SI.
Vergleiche: ASN, COS, TAN

Wirkung:
Errechnet den Sinus des Winkels X.

Anmerkungen:
Diese Funktion errechnet den Sinus des Winkels X, wobei X in Alt-Grad, Radial-Wert oder Neu-Grad ausgedrückt wird, abhängig vom Modus, in den der Computer mit den Kommandos DEGREE, RADIAN oder GRAD gesetzt wird.

Beispiel:
```
1Ø: DEGREE
2Ø: PRINT "SIN OF 3Ø IS " ; SIN (3Ø)
3Ø: PRINT "SIN OF 9Ø IS " ; SIN (9Ø)
4Ø: END

RUN
SIN OF 3Ø IS Ø.5
SIN OF 9Ø IS 1.
```

SORGN C,V,Gp

FORMAT: 1. SORGN

Abkürzung: SO.
Vergleiche: GLCURSOR

Wirkung:
SORGN wird benutzt, um den Ursprungspunkt der Koordinaten zum Zeichnen mit dem Zeichenstift zu verändern.

Anmerkungen:
Dieser Befehl ist nur effektiv im Grafik-Modus und wird benutzt, um die jetzige Position des Zeichenstifts als neuen Ursprungspunkt der Koordinaten anzugeben.
Das Zeichnen einer Figur ist nicht leicht für den Drucker, wenn der Ursprungspunkt der Koordinaten am linken Papierrand liegt. In diesem Fall, bewegen Sie den Zeichenstift bis an einen beliebigen Punkt auf dem Blatt mit Hilfe von GLCURSOR und geben diese Position als Ursprungspunkt der Koordinaten mit Hilfe von SORGN an.
Dies erleichert das Zeichnen einer Figur, wenn die jetzige Position des Zeichenstifts als Referenzpunkt genommen wird.

Beispiel:
1Ø: GRAPH
2Ø: GLCURSOR (6Ø, 4Ø)
3Ø: SORGN

[3Ø] Gibt die jetzige Position (X = 6Ø, Y = 4Ø) des Zeichenstifts als neuen Ursprungspunkt der Koordinaten an.

SQR

C,V,F

FORMAT: 1. SQR(X)

Abkürzung: SQ.
Vergleiche:

Wirkung:
Errechnet die Quadratwurzel des Ausdrucks X.

Anmerkungen:
Wenn der Ausdruck eine negative Zahl erzeugt, erhalten Sie mit SQR(X) einen Fehlercode. Anstalle von SQR kann $\sqrt{\ }$ verwendet werden.

Beispiel:
SQR(5)
2.236Ø67977

STOP V

FORMAT: 1. STOP

Abkürzung: S.
Vergleiche: END, CONT

Wirkung:
STOP wird benutzt, um die Ausführung eines Programms für Diagnosezwecke zu stoppen.

Anmerkungen:
Wenn STOP während des Programmablaufs angetroffen wird, stoppt der PC-1360 die Ausführung und es erscheint eine Meldung wie 'BREAK IN 2ØØ', wo 2ØØ die Nummer der Zeile ist, die STOP enthält. STOP wird während der Entwicklung eines Programms benutzt, um den Ablauf des Programms zu kontrollieren oder den Zustand der Variablen zu überprüfen. Sie können die Ausführung mit dem CONT-Kommando fortsetzen.

Beispiel:
1Ø: STOP

Läßt die Meldung 'BREAK IN 1Ø' auf der Anzeige erscheinen.

STR$ C,V,F

FORMAT: 1. STR$ (numerischer wert)

Abkürzung: STR.
Vergleiche: VAL

Wirkung:
Wandelt numerische Daten in Zeichenfolgendaten um.

Anmerkungen:
Die STR$-Funktion wandelt eine ganze Zahl in eine Zeichenfolge um. Die Zeichenfolge besteht aus den gleichen Ziffern, wie die ursprüngliche Zahl, wird aber als eine Zeichenfolge von SHARP ASCII-Zeichen in der weiteren Verarbeitung behandelt. Die STR$-Funktion hat den entgegengesetzten Effekt von der VAL-Funktion.

Sind die numerischen Daten negativ, so steht vor der Zeichenfolge ein Minuszeichen (−). Ist der numerische Wert zu groß, um in einer Zeichenfolgenvariablen enthalten zu sein, wird er in der Gleitpunktnotation angegeben.

Beispiel:

```
.
.
.
.
11Ø: N = N *3
12Ø: A$ = STR$ (N)
13Ø: B$ = LEFT$ (A$, 1)
14Ø: M = VAL (B$)
.
.
.
.
```

[11Ø] Das Programm führt einige Berechnungen auf der numerischen Variablen N aus.

[12Ø] Die numerische Variable N wird in die Zeichenfolgen-Variable A$ umgewandelt. Zeichenfolgen-Variablen können viel leichter manipuliert werden als numerische Zeichen. In diesem Beispiel nehmen wir an, daß die erste Ziffer der Zahl benötigt wird. Vielleicht ist sie ein Code für einen weiteren Prozeß. Da wir die Zahl in eine Zeichenfolge umgewandelt haben, können wir irgendeines der Zeichenfolgen-Manipulationskommandos wie LEFT$, RIGHT$, MID$ benutzen.

[13Ø] Speichert nur die erste Ziffer der Zahl, oder das Zeichen wie es jetzt vom Programm behandelt wird, in der Zeichenfolgen-Variablen B$.

[14Ø] Die Einzelziffer wird wieder in eine numerische Variable umgewandelt, so daß sie vom Programm als eine Zahl behandelt werden kann.

TAN

C,V,F

FORMAT: 1. TAN(X)

Abkürzung: TA.
Vergleiche: ATN, COS, SIN

Wirkung:
Errechnet den Tangens des Winkels X.

Anmerkungen:
Diese Funktion errechnet den Tangens des Winkels X, wo in Alt-Grad, Radial-Wert oder Neu-Grad angegeben wird, abhängig vom Modus, in welchen der Computer mit Hilfe des Kommandos DEGREE, RADIAN oder GRAD gesetzt wird. Ist X gleich 90°, erhalten Sie eine Fehlermeldung.

Beispiel:
```
1Ø: DEGREE
2Ø: PRINT "TAN OF Ø IS"; TAN (Ø)
3Ø: PRINT "TAN OF 45 IS"; TAN (45)
4Ø: END

RUN
TAN OF Ø IS Ø.
TAN OF 45 IS 1.
<
```

TEXT

C

FORMAT: 1. TEXT

Abkürzung: TE.
Vergleiche: BASIC

Wirkung:
Setzt den Rechner in den Text-Modus.
(Nur erlaubt im Programm-Modus.)

Anmerkungen:
Die TEXT-Funktion wird gebraucht, wenn ein Programm eingegeben werden soll, das für einen höheren Computer geschrieben wurde. Die Eingabe durch den PC-1360 wird an den Personalcomputer über das serielle E/A-Interface gesendet.

Die Ausführung des TEXT-Kommandos schaltet den Rechner in den Text-Modus. Im Text-Modus werden eine Zahl, die der Zeilennummer entspricht, und dann die Daten, die den Programmkommandos oder Daten entsprechen, eingegeben. Dann drücken Sie die ENTER-Taste, um die Eingabe in den Programm-/Datenspeicherbereich zu schreiben.
Anders als im BASIC werden die geschriebenen Inhalte aber nicht in Kommandos (interne Codes) umgewandelt. Der Text wird in der Form von Zeichen und/oder Zahlen im ASCII-Code gespeichert. Der Text wird in der Reihenfolge der Zeilennummer am Anfang jeder Zeile sortiert. (Zeilennummer-Aufbereitungsfunktion.)

Der im Text-Modus eingegebene Text wird gespeichert wie er ist. Deshalb werden BASIC-Kommandoabkürzungen (wie I. für INPUT), so wie sie eingegeben werden, angezeigt und gespeichert.

Ein Programm, das im internen Code des PC-1360 mit gesetztem Text-Modus gespeichert ist, wird in den ASCII-Code umgewandelt.

Während der Programmumwandlung wird in der 4. Zeile am rechten Rand der Anzeige ein "**" gezeigt.

Das Aufforderungszeichen des Text-Modus ist "<". (Normalerweise ist es ":>".)

Eine Zeile im Text-Modus (einschließlich der Zeilennummer und dem **ENTER**) darf nicht länger als 80 Zeichen (80 Byte) sein. Ist die Zeile länger als 80 Zeichen wegen der Programmumwandlung, wird der überzählige Teil gelöscht.

80 Byte (das Ende ist ein **ENTER**)

10: PRINT "ABC ..... 1234567890"

Umgewandelt in den ASCII-Code.

10: PRINT "ABC 1234567

80 Zeichen (das Ende ist ein **ENTER**)

In diesem Beispiel ist das PRINT-Kommando zwei Byte im internen Code, beansprucht aber 6 Byte im ASCII-Code. Deshalb werden die letzten paar Zeichen gelöscht ('890").

Die Anzahl von Byte nimmt zu, wenn ein Programm vom internen Code in den ASCII-Code umgewandelt wird, wie im Beispiel ersichtlich. Wenn dabei die Kapazität des Programmbereichs überschritten wird, wird das bis zu diesem Punkt umgewandelte Programm wieder in den internen Code umgewandelt und Sie erhalten eine Fehlermeldung (ERROR 6).

Wurde ein Passwort gesetzt, erhalten Sie eine Fehlermeldung (ERROR 1), wenn Sie das TEXT-Kommando ausführen.

TROFF C,V

FORMAT: 1. TROFF

Abkürzung: TROF.
Vergleiche: TRON

Wirkung:
TROFF wird benutzt, um den Trace-Modus auszuschalten.

Anmerkungen:
Die Ausführung von TROFF hat zur Folge, daß das Programm wieder auf normale Weise ausgeführt wird.

Beispiele:
1Ø: TRON
2Ø: FOR I = 1 TO 3
3Ø: NECT I
4Ø: TROFF

Dieses Programm bei seiner Ausführung die Zeilennummern 1Ø, 2Ø, 3Ø, 3Ø, 3Ø, und 4Ø an, wenn Sie die ↓-Taste drücken.

TRON

C,V

FORMAT: 1. TRON

Abkürzung: TR.
Vergleiche: TROFF

Wirkung:
TRON wird benutzt, um den Trace-Modus einzuschalten.

Anmerkungen:
Der Trace-Modus unterstützt die Fehlersuche in Programmen. Ist der Trace-Modus eingeschaltet, so wird die Zeilennummer der gerade ausgeführten Zeile auf der Anzeige ausgegeben. Der PC-1360 wartet dann auf den Abwärts-Pfeil, um die nächste Zeile auszuführen. Mit dem Aufwärts-Pfeil kann die gerade ausgeführte Zeile zur Anzeige gebracht werden. Der Trace-Modus bleibt so lange eingeschaltet, bis er mit einem TROFF-Befehl wieder ausgeschaltet wird oder bis Sie die Tastenfolge **SHIFT** und CA/**CLS** eingeben. Nachdem ein Ergebnis im Trace-Modus an der im CURSOR-Kommando spezifizierten Stelle im Display angezeigt wurde, erscheint in der darauffolgenden Zeile die nächste Zeilennummer.

Wenn im Trace-Modus nach einem CURSOR-Kommando die Startposition spezifiziert wurde, wird diese gelöscht, wenn Variablen aufgerufen oder Berechnungen manuell durchgeführt werden.

Beispiele:

```
1Ø: TRON
2Ø: FOR I = 1 TO 3
3Ø: NEXT I
4Ø: TROFF
```

Dieses Programm gibt bei seiner Ausführung die Zeilennummern 1Ø, 2Ø, 3Ø, 3Ø, 3Ø und 4Ø an, wenn Sie die ↓-Taste drücken.

USING C,V

FORMAT: 1. USING
2. USING "aufbereitung spezifikation"
3. USING zeichen variable

Abkürzung: U.
Vergleiche: LPRINT, MOF, PAUSE, PRINT
Weitere Erklärung zu USING finden Sie in Anhang C.

Wirkung:
USING wird gebraucht, um die angezeigte oder gedruckte Ausgabe zu formatieren.

Anmerkungen:
USING kann einzeln oder als Erweiterung eines LPRINT-, PAUSE- oder PRINT-Befehls eingesetzt werden. USING erstellt eine Ausgabeformatierung, die für alle folgenden Ausgaben Gültigkeit hat, bis die Formatierung durch ein neues USING geändert wird.

Die Aufbereitungs-Spezifikation von USING besteht aus einer in Anführungszeichen gesetzten Zeichenfolge, die aus folgenden Zeichen zusammengesetzt ist:

#	Rechtsbündiges Feld für numerische Zeichen
.	Dezimalpunkt
^	Gibt an, daß die Zahlen in Exponentialschreibweise wiedergegeben werden
&	Linksbündiges Feld für alphanumerische Zeichen.

Die Spezikation "# # # #" erstellt z.B. ein rechtsbündiges Feld für numerische Zeichen mit Platz für 3 Ziffern und das Vorzeichen. In numerischen Feldern muß immer eine Position für das Vorzeichen vorgesehen sein, auch wenn nur positive Zahlen ausgegeben werden sollen.

Aufbereitungs-Spezifikationen können mehr als ein Feld beschreiben. "# # # #&&&&" setzt z.B. ein numerisches Feld und ein alphanumerisches Feld direkt nebeneinander.

Wird keine Spezifikation gegeben, wie im Format 1, wird die Spezialformatierung abgeschaltet.

Beispiele:

	Anzeige
10: A = 125: X$ = "ABCDEF"	
20: PRINT USING "##.##^ „;A	**1.25E 02**
30: PRINT USING "&&&&&&&&";X$	**ABCDEF**
40: PRINT USING "####&&&";A;X$	**125ABC**

VAL

C,V,F

FORMAT: 1. VAL (X$)
2. VAL("string")

Abkürzung: V.
Vergleiche: STR$

Wirkung:
Wandelt eine Zeichenfolge von numerischen Zeichen in einen Dezimalwert um.

Anmerkungen:
VAL ist die umgekehrte Funktion der STR$-Funktion. Sie ändert eine Zeichenfolge, die aus numerischen Zeichen besteht, in einen numerischen Wert, der als Zahl in der weiteren Verarbeitung benutzt werden kann.

Ist die Zeichenfolge dezimal, muß sie aus den Zeichen Ø—9 bestehen, und einen wahlweisen Dezimalpunkt und ein Vorzeichen haben. In dieser Form ist VAL das Gegenteil der STR$-Funktion.
Werden unzulässige Zeichen eingegeben, wird die Umwandlung bis zum ersten unzulässigen Zeichen durchgeführt.

Beispiel:

```
1Ø: INPUT "CYCLE FREQUENCY ";A$
15: IF ASC(A$)<48 OR ASC (A$)>57 THEN 1ØØ
2Ø: F = VAL (A$)
3Ø: PRINT F
4Ø: STOP
 .
 .
 .
 .
1ØØ: PRINT "MUST BE A NUMBER" : GOTO 1Ø
```

[1Ø] Gibt eine Zeichenfolge ein, die in einen numerischen Wert umgewandelt werden soll.
[2Ø] Die Zeichenfolge wird in ihr numerisches Äquivalent umgewandelt. Die Eingabe einer Zahl als Zeichenfolge gibt dem Programmierer die Möglichkeit, die Eingabe zu überprüfen, um sicherzugehen, daß der Typ richtig ist oder die Eingabe in der richtigen Reihenfolge im Programm selbst vorgenommen wurde.

WAIT C,V

FORMAT: 1. WAIT ausdruck
2. WAIT

Abkürzung: W.
Vergleiche: PAUSE, PTINT

Wirkung:
WAIT wird benutzt, um die Zeit festzusetzen, die die Ausgabe einer Information auf der Anzeige während des Programmablaufs wiedergegeben werden soll.

Anmerkungen:
Bei normaler Programmausführung wartet der PC-1360 nach einem PRINT-, GPRINT-, PSET-, PRESET-, LINE-Kommando, bis die **ENTER**-Taste gedrückt wird. WAIT weist den PC-1360 an, eine Ausgabe für eine vorgegebene Zeit anzuzeigen und dann die Ausführung des Programms automatisch fortzusetzen (ähnlich wie bei PAUSE). Der ausdruck nach WAIT setzt die Länge des Intervalls fest. Der Intervall kann jeden Wert von 0 bis 65535 haben. Dabei bedeutet jede Erhöhung um 1 eine Verlängerung um 1/59 Sekunde. WAIT 0 ist zu schnell, als daß die Ausgabe gut gelesen werden könnte; WAIT 65535 ist etwa 19 Minuten. WAIT ohne Ausdruck setzt den PC-1360 in seinen ursprünglichen Wartezustand, bis die **ENTER**-Taste gedrückt wird.

Beispiele:
1Ø: WAIT 59

Der Rechner wartet nach PRINT etwa 1 Sekunde.

KAPITEL 10 PANNENHILFE

In diesem Kapitel sollen Sie einige Hinweise erhalten, was Sie unternehmen können, wenn Ihr SHARP PC-1360 nicht tut, was Sie von ihm erwarten. Es ist in zwei Teile gegliedert - der erste beschäftigt sich mit allgemeiner Bedienung des Gerätes, und der zweite mit der BASIC-Programmierung. Für jedes Problem schlagen wir eine Reihe von Lösungen vor. Sie sollten jede von ihnen versuchen, aber nur eine zur Zeit, bis Sie das Problem gelöst haben.

Bedienung des Gerätes

Wenn:	Dann sollten Sie:
Wenn Sie das Gerät einschalten, aber nichts in der Anzeige erscheint.	1. Überprüfen. ob der Gleitschalter in der ON-Stellung ist. 2. Die ON BRK -Taste betätigen, um festzustellen, ob sich das Gerät automatisch abgeschaltet hat. 3. Die Batterien wechseln. 4. Den Kontrast einstellen.
Die Anzeige zwar funktioniert, aber keine Reaktion auf Tastendruck erfolgt.	1. CLS drücken, um die Anzeige zu löschen. 2. CA (SHIFT CLS) drücken, um die Anzeige zu löschen. 3. Das Gerät aus- und dann wieder einschalten. 4. Irgendeine Taste drücken, festhalten und den ALL RESET-Schalter drücken. 5. Den ALL RESET-Schalter ohne zusätzlichen Tastendruck betätigen.

Sie eine Rechnung oder Antwort eingeben, aber keine Reaktion erfolgt.	1. Drücken Sie **ENTER**.
Sie ein BASIC-Programm abarbeiten lassen, etwas angezeigt wird und das Programm dann anhält.	1. Drücken Sie **ENTER**.
Sie eine Rechnung eingeben und diese in der Form einer BASIC-Anweisung angezeigt wird (Doppelpunkt nach der ersten Zahl).	1. Schalten Sie vom PROgramm- in den RUN-Modus um.
Sie keinerlei Reaktion auf Tastenbetätigung erhalten.	1. Halten Sie irgendeine Taste fest und drücken Sie den ALL RESET-Schalter. 2. Wenn Sie dann noch immer keine Reaktion erhalten, drücken Sie den ALL RESET-Schalter, ohne dabei eine andere Taste festzuhalten. Dabei werden jedoch alle Daten, Programme und Speicherinhalte gelöscht.

Fehlersuche im BASIC

Wenn Sie ein neues BASIC-Programm eingeben, wird dieses in der Regel beim ersten Startversuch nicht laufen. Selbst, wenn Sie nur ein Programm abtippen, von dem Sie wissen, daß es korrekt ist, wie z.B. die in diesem Handbuch vorgestellten, dürfte Ihnen normalerweise mindestens ein Tippfehler unterlaufen. Handelt es sich um ein längeres Programm, wird es oft auch mindestens einen logischen Fehler enthalten. Es folgen eingige grundsätzliche Hinweise, wie Sie solche Fehler finden und korrigieren.

Sie starten Ihr Programm und erhalten eine Fehlermeldung:

1. Schalten Sie zurück in den PROgramm-Modus und benutzen Sie die ↑- oder ↓-Taste, um die fehlerhafte Zeile ins Display zu rufen. Der Cursor wird sich an der Stelle befinden, wo der PC-1360 verwirrt wurde.

2. Wenn Sie aus der Art, wie die Programmzeile geschrieben ist, keinen offensichtlichen Fehler entnehmen können, kann das problem auch an den von Ihnen verwendeten Werten liegen. So wird beispielsweise CHR$(A) einen Leerraum erzeugen, wenn A den Wert 1 hat. Überprüfen Sie die Werte der von Ihnen verwendeten Variablen, indem Sie entweder im RUN- oder im PROgramm-Modus die einzelnen Variablennamen gefolgt von **ENTER** eingeben.

Sie starten das Programm mit RUN und erhalten keine Fehlermeldung, doch das Programm tut nicht, was Sie von ihm erwärten:

3. Überprüfen Sie das Programm Zeile für Zeile unter Verwendung von LIST und der ↓- und ↑-Tasten, um herauszufinden, ob Sie das Programm korrekt eingegeben haben. Es ist erstaunlich, wieviele Fehler beim bloßen erneuten Durchsehen eines Programms gefunden werden können!

4. Versuchen Sie, jede einzelne Zeile beim Durchsehen so zu interpretieren, als wären Sie ein Computer. Nehmen Sie einfache Werte und realisieren Sie die Operationen der einzelnen Zeilen, um herauszufinden, ob Sie die gewünschten Ergebnisse erhalten.

5. Fügen Sie eine oder mehrere zusätzliche PRINT-Anweisungen in das Programm ein, um die einzelnen Werte und Tastenbelegungen zur Anzeige zu bringen. Benutzen Sie diese, um die korrekten Teile des Programms von den möglicherweise fehlerhaften zu isolieren. Diese Vorgehensweise ist auch nützlich um zu bestimmen, welche Teile des Programms schon abgearbeitet wurden. Sie können den Programmablauf auch an kritischen Stellen vorübergehend mit STOP unterbrechen, um dann einzelne Variablen zu überprüfen.

6. Verwenden Sie TRON und TROFF, entweder als Befehle oder als Programmbestandteile, um den Programmablauf durch die einzelnen Zeilen hindurch verfolgen zu können. Halten Sie das Programm an kritischen Punkten an, um den Inhalt von wichtigen Variablen zu überprüfen. Dies ist zwar eine sehr langsame Methode, Fehler aufzuspüren, aber es ist manchmal die einzige.

Um das Programm fortzusetzen, drücken Sie einmal die ↓-(Abwärtspfeil)Taste. Daraufhin wird die nächste Programmzeile ausgeführt und ihre Zeilennummer angezeigt. Um die Programmzeile zu überprüfen, können Sie dann die Aufwärtspfeil-Taste betätigen. Sie können auch die Inhalte der einzelnen Variablen überprüfen, indem Sie den Variablennamen eingeben und ENTER drücken:

A ENTER (wenn A = 4 vor der Operation A ENTER eingegeben wird)

```
                                                    4.
```

Es ist notwendig, zur Abarbeitung jeder einzelnen Programmzeile die ↓-(Abwärtspfeil)Taste zu drücken. Wenn Sie den normalen Ablauf von Zeile zu Zeile unterbrechen wollen, drücken Sie die ENTER-Taste, um den Programmablauf zu unterbrechen. Wenn Sie es sich anders überlegt haben und das Programm fortführen wollen, geben Sie das CONT-Kommando.

Ein Beispiel unseres Hypothenuse-Programms:

Eingabe	Anzeige
	<
T R O N	TRON_
ENTER	<
R U N	RUN_
ENTER	?
3	3_
ENTER	?
4	4_
ENTER	11Ø:
↑	1Ø: INPUT A,B
↓	2Ø:
↑	2Ø: A=A∗A : B=B∗B
A	A_
ENTER	9.
B	B_
ENTER	16.
↓	3Ø:
H	H_
ENTER	5.
↓	HYPOTENUSE = 5.
↑	4Ø: PRINT"HYPOTENUSE = "; H
↓	4Ø:
↓	>

Unabhängig davon, wie vorsichtig Sie auch programmieren, werden Sie ein Programm schreiben, welches nicht das macht, was sie von ihm erwarten. Dann geht es darum, den Fehler zu finden. SHARPs Entwickler haben deshalb eine spezielle Programmablauf-Routine, genannt "Trace"-Modus, vorbereitet. Im Trace-Modus zeigt der PC-1360 die Zeilennummer jeder Programmzeile an und stoppt nach der Ausführung jeder einzelnen Zeile. Das gibt Ihnen die Möglichkeit, die Reihenfolge der Instruktionen bei ihrer Ausführung zu verfolgen. Wenn das Programm nach der Ausführung einer Zeile anhält, können Sie die Werte der Variablen überprüfen oder ändern.

Die Einschaltung des Trace-Modus ist ganz einfach: Durch die Eingabe von TRON. Die TRON-Anweisung kann als Kommando (im RUN-Modus) oder auch innerhalb eines Programms als Befehl verwendet werden. Wird TRON als Kommando verwendet, informiert es den PC-1360, daß eine Ablaufverfolgung während der Ausführung aller nachfolgender Programme benötigt wird. Die zu verfolgenden Programme werden dann ganz normal gestartet, mit Hilfe eines GOTO- oder RUN-Kommandos.

Wird TRON innerhalb des Programms verwendet, so wird der Trace-Modus erst beim Erreichen der entsprechenden Zeile eingeschaltet. Wenn aus irgendeinem Grund diese Zeile nie erreicht wird, bleibt der Trace-Modus ineffektiv. Nach dem Einschalten bleibt der Trace-Modus effektiv, bis er von einem TROFF-Befehl ausgeschaltet wird. TROFF kann ebenso wie TRON als Kommando oder Anweisung verwendet werden. Den Trace-Modus kann man außerdem durch die folgende Tastenfolge ausschalten:

CA
SHIFT **CLS**

Um ein Beispiel für die Benutzung des Trace-Modus zu haben, geben Sie bitte das folgende Programm ein, welches die Länge der Hypothenuse in einem rechtwinkligen Dreieck bei gegebener Länge der beiden anderen Seiten bestimmt:

PROGRAMMLISTE

```
1Ø:INPUT A, B
2Ø: A=A*A : B=B*
3Ø: H=√ (A+B)
4Ø: PRINT "HYPOTENUSE=";H
```

Im RUN-Modus, geben Sie das TRON-Kommando gefolgt vom RUN-Kommando. Beachten Sie, daß der INPUT-Befehl wie gewohnt abgearbeitet wird, indem ein Fragezeichen für jeden eingegebenen Wert angezeigt wird. Sobald Sie zwei Werte eingegeben haben, erscheint die Zielennummer der INPUT-Anweisung:

```
10:
```

Wenn Sie die ↑-(Aufwärtspfeil)Taste drücken und festhalten, können Sie die ganze Zeile überprüfen:

```
10: INPUT A,B
```

Trace-Modus wird das errechnete Ergebnis an der Stelle, die durch ein CURSOR-Kommando spezifiziert wurde, ausgegeben und die Zeilennummber wird in der darauffolgenden Zeile angezeigt.

Werden nun manuell im Trace-Modus Variablen abgerufen oder Rechnungen durchgeführt, während die Anzeige-Startposition mit dem Cursor-Befehl vorgegeben worden ist, so wird die Anzeige-Startposition gelöscht.

KAPITEL 11 INSTANDHALTUNG DES PC-1360

Um reibungsloses Funktionieren Ihres PC-1360 zu gewährleisten, empfehlen wir die Beachtung der folgenden Punkte:

- Gehen Sie immer vorsichtig mit dem Computer um, da die Flüssigkristallanzeige aus Glas gefertigt ist.

- Halten Sie den Computer von starken Temperaturschwankungen fern, ebenso von Feuchtigkeit oder Staub. Wenn Sie bei warmem Wetter Ihren PKW längere Zeit der direkten Sonne aussetzen und sich hohe Temperaturen aufbauen, kann das Ihren Computer beschädigen.

- Benutzen Sie zur Reinigung des Computers ausschließlich ein trockenes, weiches Tuch. Verwenden Sie niemals Lösungsmittel, Wasser oder ein feuchtes Tuch.

- Um das Auslaufen der Batterien zu vermeiden, entfernen Sie diese, wenn Sie den Computer für längere Zeit nicht benutzen wollen.

- Wenn Sie die Hilfe einer Werkstatt benötigen, geben Sie den Computer nur in ein autorisiertes SHARP Service Center.

- Wenn der Computer starker statischer Aufladung oder auch starkem Lärm ausgesetzt ist, kann er sich unter Umständen "aufhängen" (d.h. auf Tastendruck nicht mehr reagieren). Falls dies auftritt, drücken Sie den ALL RESET-Schalter und halten dabei eine Taste fest. (Siehe Pannenhilfe.)

- Bewahren Sie dieses Handbuch gut auf, falls Sie später einmal etwas nachschlagen wollen.

ANHANG A FEHLERMELDUNGEN

Es gibt neun verschiedene Fehler-Codes, die im PC-1360 eingebaut sind. Die folgende Tabelle erläutert diese Codes.

Fehlernummer **Bedeutung**

1 Syntax-Fehler
Der PC-1360 kann nicht verstehen, was Sie eingegeben haben. Prüfen Sie die Eingabe auf Dinge wie Semikolon am Ende einer PRINT-Anweisung, falsch geschriebene Wörter und fehlerhafte Anwendungen.

```
3*/2
```

2 Rechenfehler
Hier haben Sie vermutlich eines der drei folgenden Dinge getan:

1. Versucht, eine zu große Zahl zu benutzen. Das Rechenergebnis ist größer als 9.999999999E 99.
2. Versucht, durch Null zu teilen.

```
5/Ø
```

3. Versucht, eine unlogische Rechnung auszuführen.

```
LN -3Ø oder ASN 1.5
```

3 Unzulässige Funktion (DIMensionierungsfehler/Argumentfehler)
Die Feldvariable existiert bereits.
Das Feld wurde spezifiziert, aber nicht dimensioniert.
Der Index des Feldes übersteigt die in der DIM-Anweisung vorgegebene Größe.

```
DIM B(256)
```

Unzulässiges Funktionsargument. Das heißt Sie haben versucht, den Computer etwas tun zu lassen, wozu er nicht imstande ist.
Das Intervall ist größer als 65535.

```
WAIT 66ØØØ
```

4 Zu hohe Zeilennummer
Hier haben Sie vermutlich eines der folgenden zwei Dinge gemacht:
1. Versucht, mit GOTO, GOSUB, RUN, LIST oder THEN usw. eine nicht vorhandene Zeile anzusprechen.
2. Versucht, eine zu große Zeilennummer zu verwenden. Die höchstmögliche Zeilennummer ist 65279.

5 NEXT ohne FOR ...
Das Einrichten einer Subroutine übersteigt 1Ø Stufen. Das Einrichten einer FOR-Schleife übersteigt 5 Stufen. Die Befehle RETURN ohne GOSUB, NEXT ohne FOR, oder READ ohne DATA.

Fassungesvermögen des Puffers überschritten.

6 Speicherkapazität überschritten.
Diese Fehlermeldung erhalten Sie normalerweise, wenn Sie versuchen, ein Feld zu DIMensionieren, das zu groß für die Speicherkapazität ist. Sie kann aber auch auftreten, wenn ein Programm zu lang ist, oder wenn der Reservespeicher mehr als 144 Byte enthält.

7 PRINT USING-Fehler
Das bedeutet, daß in einer USING-Anweisung eine unzulässige Formatspezifikation enthalten ist.

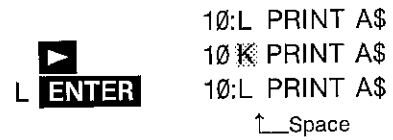
8 E/A-Anschluß-Fehler
Dieser Fehler kann nur auftreten, wenn Sie den zusätzlichen Drucker und/oder Kassettenrecorder an den PC-1360 angeschlossen haben. Dieser Fehler kann auch auftreten, wenn Sie den seriellen E/A-Anschluß benutzen. Es bedeutet, daß Sie ein Kommunikationsproblem zwischen dem E/A-Anschluß und dem PC-1360 haben.

9 Andere Fehler
Dieser Fehlercode wird angezeigt, wenn der Computer ein Problem hat, das mit den anderen Fehlercodes nicht zu erfassen ist. Eine der häufigsten Ursachen für das Auftreten dieser Meldung ist der Versuch, Daten einer Variablen unter einem bestimmten Namen (z.B. A$) anzusprechen, während die Daten der Variablen unter einem anderen Namen (d.h. A) abgespeichert worden waren.

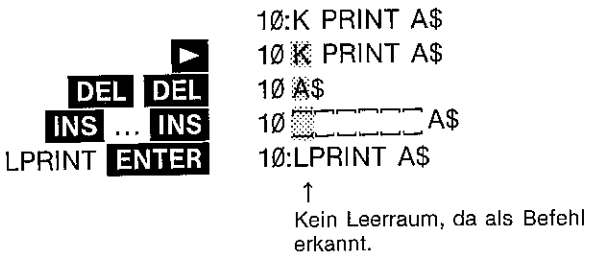
Eingabefehler

Wenn Sie ein Programm ausführen, kann eine Fehlermeldung aufgrund eines Eingabefehlers auftreten. Beachten Sie in diesem Fall die folgenden Anmerkungen:

Beispiel: Wenn KPRINT anstelle von LPRINT eingegeben wird.

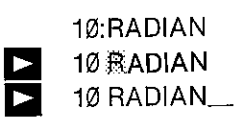


Wenn Sie auf diese Weise die Korrektur durchführen, versteht der Computer sie nicht als Befehl. In diesem Beispiel müssen Sie KPRINT löschen und dann LPRINT neu eingeben.



Mit der Cursor-Taste können Sie den Befehl auf korrekte Eingabe überprüfen.

(Richtige Eingabe)



(Falsche Eingabe)

ANHANG B ZEICHENCODE-TABELLE

Die folgende Tabelle zeigt die Umwandlungswerte für den Einsatz von CHR$ und ASC. Die Spalte zeigt das erste hexadezimale Zeichen oder die ersten vier Binärbits, die Reihe zeigt das zweite hexadezimale Zeichen oder die zweiten Binärbits. Die obere linke Ecke jedes Kastens enthält die Dezimalzahl für das Zeichen. Die untere rechte Ecke zeigt das Zeichen. Wird kein Zeichen angegeben, bedeutet es, daß das Zeichen unzulässig ist auf dem PC-1360.

Zum Beispiel, das Zeichen "A" ist dezimal 65 oder hexadezimal 41 oder binär Ø1ØØØØØ1. Das Zeichen '√‾ ' ist dezimal 252 oder hexadezimal FC oder binär 11111ØØ.

Die Zeichencodes werden folgendermaßen angezeigt:
Beispiele:
Code für "*"
Hexadezimal: &2A
Dezimal: 42 (32 + 1Ø)

Code für "√‾ "
Hexadezimal: &FC
Dezimal: 252 (24Ø + 12)

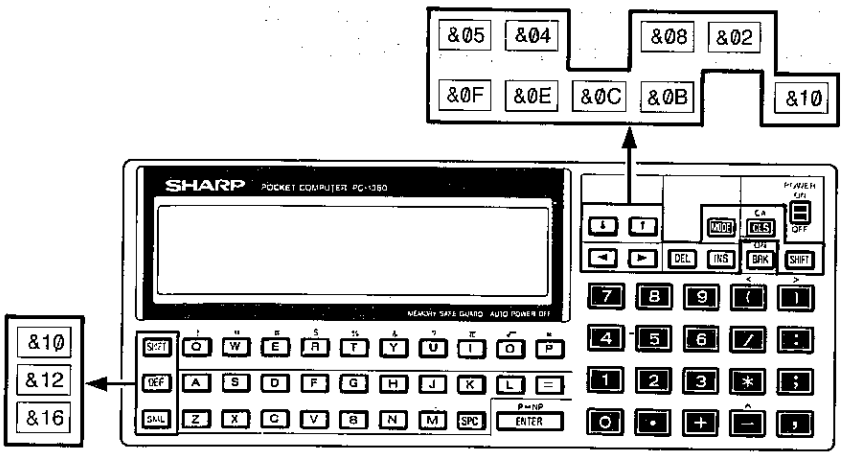
Anzeige eines Zeichens mit Hilfe des CHR$-Kommandos:
Das Zeichen für den Code 0 (&00) in der Tabelle ist leer. Es wird nichts angezeigt. Die Vierecke in der Tabelle, wo keine Zeichen angegeben sind, bedeuten Leerzeichen.

Drucken eines Zeichens auf dem CE-126P mit Hilfe des CHR$-Kommandos:
- Benuzten Sie nicht den Code Ø (&ØØ).
- Die Vierecke in der Tabelle, wo keine Zeichen angegeben sind, bedeuten Leerzeichen.
- Die Codes 249 (&F9) und 25Ø (&FA) sind Leerzeichen.
- Das Zeichen für den Code 92 (&5C) ist "◣".

Drucken eines Zeichens auf dem CE-140P mit Hilfe des CHR$-Komandos:

- Für das Drucken von Zeichen auf dem CE-140P sind die Codes 8 (&Ø8), 1Ø (&ØA), 11 (&Ø8), 13 (&ØD),und 27 (&1B) die Steuercodes für den Drucker. Die übrigen, d.h. Ø (&ØØ) bis 31 (&1F) sind NULL.
- Alle anderen als die Ø (&ØØ) bis 31 (&1F) Codes, die nicht für Zeichen gebraucht werden, werden als Leerzeichen gedruckt.
- Die Codes 249 (&F9) und 250 (&FA) sind Leerzeichen.

Die Abbildung hier unten zeigt die Tastenanordnung und die Werte, die angezeigt werden, wenn INKEY$ eingesetzt wird. Die meisten Codes sind in der Tastencode-Tabelle angegeben, aber diese Abbildung zeigt auch die nicht ausdruckbaren Tastencodes.



INKEY$ ZEICHENCODE-TABELLE

High / Low	0	1	2	3	4	5	6
0		SHIFT	SPC	0		P	
1				1	A	Q	
2	CLS	DEF		2	B	R	
3				3	C	S	
4	↑			4	D	T	
5	↓			5	E	U	
6		SML		6	F	V	
7				7	G	W	
8	MODE		(	8	H	X	
9			)	9	I	Y	
A			*	:	J	Z	
B	INS		+	;	K		
C	DEL		,		L		
D	ENTER		−	=	M		
E	►		.		N		
F	◄		/		O		

Ersten 4 Bits

Zweiten 4 Bits

Hex. /Binär	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	E 1110	F 1111
0 0000	0 NUL	16	32 SPACE	48 0	64 @	80 P	96 `	112 p	128	224	240
1 0001	1	17	33 !	49 1	65 A	81 Q	97 a	113 q	129	225	241
2 0010	2	18	34 "	50 2	66 B	82 R	98 b	114 r	130	226	242
3 0011	3	19	35 #	51 3	67 C	83 S	99 c	115 s	131	227	243
4 0100	4	20	36 $	52 4	68 D	84 T	100 d	116 t	132	228	244
5 0101	5	21	37 %	53 5	69 E	85 U	101 e	117 u	133	229	245 ♠
6 0110	6	22	38 &	54 6	70 F	86 V	102 f	118 v	134	230	246 ♥
7 0111	7	23	39 '	55 7	71 G	87 W	103 g	119 w	135	231	247 ♦
8 1000	8	24	40 (	56 8	72 H	88 X	104 h	120 x	136	232	248 ♣
9 1001	9	25	41)	57 9	73 I	89 Y	105 i	121 y	137	233	249 ▮
A 1010	10	26	42 *	58 :	74 J	90 Z	106 j	122 z	138	234	250 □
B 1011	11	27	43 +	59 ;	75 K	91 [	107 k	123 {	139	235	251 π
C 1100	12	28	44 ,	60 <	76 L	92 \	108 l	124 ¦	140	236	252 √
D 1101	13	29	45 –	61 =	77 M	93]	109 m	125 }	141	237	253
E 1110	14	30	46 .	62 >	78 N	94 ^	110 n	126 ~	142	238	254
F 1111	15	31	47 /	63 ?	79 O	95 _	111 o	127	143	239	255

ANHANG C FORMATIERUNG DER DATENAUSGABE

Es ist manchmal wichtig oder nützlich, neben dem Inhalt ausgegebener Daten auch das Format zu kontrollieren. Der PC-1360 bedient sich hierzu des Befehls USING. Mit Hilfe dieses Befehls können Sie spezifizieren:

- Die Anzahl der Stellen
- Die Position des Dezimalpunktes
- Wissenschaftliche Notation
- Anzahl der Zeichen in einer Zeichenfolge

Diese verschiedenen Formate bestimmen Sie mit einer "Ausgabe-Maske", die aus einer Zeichenfolgen-Konstanten oder auch einer Zeichenfolgen-Variablen bestehen kann.

```
1Ø: USING"# # # #"
2Ø: M$="&&&&&&"
3Ø: USING M$
```

Wird der Befehl USING ohne Maske benutzt, werden damit alle Spezialformate aufgehoben.

```
4Ø: USING
```

USING kann auch in einer PRINT-Anweisung benutzt werden.

```
5Ø: PRINT USING M$; N
```

Wann immer USING benutzt wird, bewirkt es die Kontrolle aller ausgegebenen Daten, bis ein neuer USING-Befehl auftritt.

Numerische Masken
Eine numerische USING-Maske darf nur benutzt werden, um numerische Werte, d.h. numerische Konstanten oder numerische Variablen, zu kontrollieren. Wird eine Zeichenfolgen-Konstante oder -Variable angezeigt, während eine numerische USING-Maske wirksam ist, wird die Maske hierauf nicht angewendet werden. Ein auszugebender Wert muß immer in den von der Maske vorgesehenen Freiraum passen. Es muß auch Raum für das Vorzeichen vorgesehen sein, selbst, wenn die Zahl immer positiv sein wird. Eine Maske, die vier Stellen zuläßt, kann also nur für dreistellige Zahlen verwendet werden.

Sie können die MDF-Funktion zusammen mit USING benutzen, um Dezimalzahlen vor dem Ausdrucken zu runden. Mit MDF können Werte im Computer auf die gleiche Anzahl von Dezimalstellen wie in der USING-Anweisung spezifiziert beibehalten werden. Normalerweise werden Werte bis zur nächsten ganzen Zahl aufgerundet, sofern der abgeschnitene Teil gleich 5 oder größer ist. MDF erlaubt Ihnen, den Schwellen-Aufrundungswert zu verändern.

Bestimmung der Anzahl von Stellen

Die gewünschte Anzahl von Stellen wird mit Hilfe des '#'-Zeichens festgelegt. Die Anzeige oder der Ausdruck beinhalten immer soviele Zeichen, wie in der Maske vorgesehen sind. Dabei erscheint die Zahl rechts in diesem Bereich, die restlichen Stellen werden durch Leerschritte aufgefüllt. Bei positiven Zahlen wird sich daher auf der linken Seite der Anzeige immer mindestens ein Leerschritt befinden. Da der PC-1360 nicht mehr als 10 Stellen erfassen kann, sollte die größte verwendete numerische Maske maximal 11 '#'-Zeichen enthalten. Wenn die Gesamtstellenzahl des ganzzahligen Teils 11 Stellen übersteigt, wird dieser ganzzahlige Teil als 11 Stellen aufgefaßt.

Anmerkung:
In allen Beispielen dieses Anhangs werden Sie am Anfang und am Ende eines Anzeigefeldes '|'-Zeichen finden, mit denen die Größe des Feldes anschaulich gemacht werden soll.

Anweisung	Anzeige
1Ø: USING "# # # #"	(Bringen Sie den PC-1360 in den RUN-Modus, geben Sie RUN ein und drücken Sie **ENTER**.)
2Ø: PRINT 25	| **25.** |
3Ø: PRINT −35Ø	|**−350.** |
4Ø: PRINT 1ØØØ	**ERROR 7 IN 40**

Beachten Sie, daß die letzte Eingabe eine Fehlermeldung hervorruft, weil 5 Stellen (4 Zahlen und eine Stelle für das Vorzeichen) benötigt werden, die Maske aber nur 4 vorsieht.

Bestimmung des Dezimalpunktes

Das Zeichen für den Dezimalpunkt ' . ' kann in einer numerischen Maske enthalten sein, um die gewünschte Position des Dezimalpunktes festzulegen. Wenn die Maske mehr Stellen bereitstellt, als für den anzuzeigenden Wert benötigt werden, werden die auf der rechten Seite übrigbleibenden Stellen mit Nullen ausgefüllt. Enthält der anzuzeigende Wert mehr Stellen, als die Maske vorsieht, werden diese abgeschnitten (nicht gerundet).

Anweisung	Anzeige
1Ø: USING "# # # #.# #"	
2Ø: PRINT 25	| **25.ØØ** |
3Ø: PRINT −35Ø.5	| **−35Ø.5Ø** |
4Ø: PRINT2.547	| **2.54** |

Bestimmung der wissenschaftlichen Notation

Das Zeichen ' ∧ ' kann in einer Maske enthalten sein, um so anzuzeigen, daß die Zahl in wissenschaftlicher Notation ausgegeben werden soll. Die Zeichen '#' und '.' werden in der Maske benutzt, um den "charakteristischen" Teil der Zahl, d.h. den Teil, der links vom "E" steht, zu formatieren. Links vom Dezimalpunkt sollten immer 2 '#' -Zeichen stehen, um genug Raum für eine ganze Zahl und das Vorzeichen zu schaffen. Der Dezimalpunkt kann, muß aber nicht, vorgegeben werden. Rechts vom Dezimalpunkt können bis zu 9 Stellen eingerichtet werden. Nach dem charakteristischen Teil wird das Exponentiationszeichen E angezeigt, gefolgt von der Stelle für das Vorzeichen und dem 2stelligen Exponenten. Das kleinste von einer Maske vorzugebende Feld für wissenschaftliche Notation wäre demnach "# #∧", damit werden Zahlen in der Form '2E 99' dargestellt. Das größtmögliche Feld für wissenschaftliche Notation wäre ' # #.# # # # # # # # # #∧', das Zahlen wie "−1.234567890E −12" darstellen kann.

Anweisung	Anzeige
1Ø: USING "# # #.# #∧"	
2Ø: PRINT 2	| **2.ØØE ØØ** |
3Ø: PRINT −365.278	| **−3.65E Ø2** |

Bestimmung alphanumerischer Masken

Zeichenfolgen-Konstanten und Variablen werden mit Hilfe des ' & '-Zeichens ausgegeben. Jedes ' & ' gibt eine im vorgesehenen Feld anzuzeigende Stelle ein. Die Zeichenfolge wird auf der linken Seite dieses Feldes angezeigt. Ist die Zeichenfolge kürzer als der dafür eingeräumte Raum, werden die rechts verbleibenden Stellen mit Leerschritten aufgefüllt. Ist die Zeichenfolge länger als das Feld, wird sie abgeschnitten.

Anweisung	Anzeige
1Ø: USING "&&&&&&"	
2Ø: PRINT"ABC"	| **ABC** |
3Ø: PRINT"ABCDEFGHI"	| **ABCDEF**|

Gemischte Masken

In den meisten Anwendungsfällen wird eine USING-Maske entweder alle notwendigen numerischen Zeichen oder alle Zeichen zur Zeichenfolgen-Formatierung beinhalten. Für bestimmte Zwecke können aber auch beide gemeinsam in einer USING-Maske enthalten sein. In solchen Fällen markiert jede Umschaltung von Zeichenfolgen formatierenden Zeichen auf Zahlen formatierende Zeichen (und umgekehrt) die Grenze für einen bestimmten Wert. So können mit einer Maske der Form "# # # # # &&&&" zwei verschiedene Werte formatiert werden—ein numerischer Ausdruck, für den 5 Stellen vorgesehen sind, und ein alphanumerischer Ausdruck, für den 4 Stellen bereitgestellt wurden.

Anweisung	Anzeige
1Ø: USING"###.##&&";25;"CR"	| **25.ØØCR** |
2Ø: PRINT–5.789;"DB"	| **–5.78DB** |

Vergessen Sie nicht:
Wurde ein USING-Format einmal spezifiziert, wirkt es sich auf alle nachfolgenden Daten aus, bis es aufgehoben oder durch einen anderen USING-Befehl ersetzt wird.

ANHANG D BEWERTUNG VON AUSDRÜCKEN UND VORRANG VON OPERATOREN

Wenn in den PC-1360 ein komplexer Ausdruck gegeben wird, bewertet er Teile dieses Ausdrucks in einer Reihenfolge, die durch die Vorrangstellung der einzelnen Teile bestimmt wird. Geben Sie den Ausdruck:

100/5 + 45

ein, entweder als Rechenoperation oder als Teil eines Programms, so weiß der PC-1360 nicht, ob Sie meinen:

$$\frac{100}{5 + 45} = 2 \qquad \text{oder} \qquad \frac{100}{5} + 45 = 65$$

Da der PC-1360 eine Möglichkeit haben muß, zwischen diesen beiden Operationen zu entscheiden, bedient er sich seiner Regeln des Operatoren-Vorrangs. Da die Division eine höhere "Priorität" hat als die Addition (siehe unten), wird er entscheiden, daß zuerst die Division durchgeführt wird und anschließend die Addition, d.h. die zweite Möglichkeit wird ausgeführt und als Ergebnis 65 ausgegeben.

Vorrang von Operatoren

Im BASIC werden Operatoren vom SHARP PC-1360 gemäß den folgenden Prioritätsregeln verarbeitet, angefangen mit der höchsten Priorität:

Stufe	Operationen
1.	Klammern
2.	Variablen und Pseudovariablen
3.	Funktionen
4.	Exponentiation (^)
5.	Einheitliches Minuszeichen, negatives Vorzeichen (−)
6.	Multiplikation und Division (*, /)

7. Addition und Subtraktion (+, −)
8. Verhältnis-Operatoren (<, <=, =, <>, >=, >)
9. Logische Operatoren (AND, OR, NOT)

Treten in einem Ausdruck zwei oder mehr Operatoren derselben Prioritätsstufe auf, wird der Ausdruck von links nach rechts verarbeitet. (Exponentiation wird von rechts nach links verarbeitet.) Beachten Sie, daß bei einer Operation A + B − C das Ergebnis dasselbe ist, ob Sie nun die Addition oder die Subtraktion zuerst ausführen.

Enthält ein Ausdruck ineinanderliegende Klammern, so wird die innerste Klammer zuerst bearbeitet, die äußerste zuletzt.

Für die Prioritätsstufen 3 und 4 gilt, daß die letzte Eingabe die höchste Priorität hat.
Zum Beispiel: $-2\wedge4 \rightarrow -(2^4)$
$3\wedge-2 \rightarrow 3^{-2}$

Beispiel für eine Bewertungsfolge

Wir gehen aus von dem Ausdruck:

```
((3+5−2)*6+2)/10^LOG 100
```

Der PC-1360 würde nun zuerst die innersten Klammern verarbeiten. Da ' + ' und ' − ' auf derselben Stufe stehen, wird von links nach rechts gerechnet, also die Addition zuerst ausgeführt:

```
((8−2)*6+2)/10^LOG 100
```

Dann würde die Subtraktion durchgeführt:

```
((6)*6+2)/10^LOG 100
```

oder:

```
(6*6+2)/10^LOG 100
```

In der nächsten Klammer würde zuerst die Multiplikation durchgeführt:

```
(36+2)/10^LOG 100
```

Und dann die Addition:

```
(38)/10^LOG 100
```

oder:

```
38/10^LOG 100
```

Nachdem nun die Klammern ausgelöst sind, hat die LOG-Funktion höchste Priorität und wird somit als nächstes ausgeführt:

```
38/10^2
```

Als nächstes folgt die Exponentiation:

```
38/100
```

Und zuletzt wird die Division ausgeführt:

```
0.38
```

Dies ist der Wert des Ausdrucks.

ANHANG E TASTENFUNKTIONEN IM BASIC

ON
BRK

(ON)
Wird benutzt, um den PC-1360 anzuschalten, wenn er sich automatisch abgeschaltet hat.
(BREAK)

- Das Betätigen dieser Taste während eines Programmablaufs bewirkt eine Unterbrechung der Programmausführung.
- Bei manuellen Operationen, Ein-/Ausgabe-Befehlen wie BEEP, CLOAD usw. wird mit Betätigung dieser Taste die Befehlsausführung unterbrochen.

SHIFT

- Die gelbe Taste mit der Aufschrift "SHIFT" muß benutzt werden, um Doppelfunktionen anzusprechen (die in brauner Schrift über jeder Taste stehenden Zeichen).
  Z.B. wird mit **SHIFT** und **U** das ? angesprochen
- Wenn etwas gespeichert oder abgerufen wird im ReSerVe-Modus, muß diese Taste vor der benannten Taste gedrückt werden.

CLS

- Wird benutzt, um Eingabe und Anzeige zu löschen.
- Wird benutzt, um Blockaden durch Fehler aufzuheben.

SHIFT **CA**

- Löscht nicht nur den Anzeigeteil, sondern initialisiert darüber hinaus den Computer.
  —Initialisierung—
  - Hebt den WAIT-Timer auf.
  - Löscht das Anzeige-Format (USING-Format).
  - Hebt den TRON-Zustand (TROFF) auf.
  - Hebt PRINT = LPRINT auf.
  - Hebt Blockade durch Fehler auf.

MODE	● Mit dieser Taste können Sie den Betriebsmodus von RUN auf PROgramm oder von PROgramm auf RUN umschalten.
SHIFT MODE	● Wird benutzt, um den Reserve-Modus auszuwählen.
0 ~ 9	● Zifferntasten
.	● Dezimalpunkt ● Wird benutzt, um eine Abkürzung eines Kommandos/ Befehls/Funktion einzugeben. ● Gibt in der Bestimmung eines USING-Formats die Stellung des Dezimalpunktes an.
E	● Wird benutzt, um in wissenschaftlicher Notation den Exponenten zu bestimmen. (Diese Taste ist eine Buchstabetaste E: Großbuchstaben.)
/	● Divisionstaste
*	● Multiplikationstaste ● Wird benutzt, um eine Feld-Variable in INPUT#, PRINT# usw. zu bestimmen.
+	● Additionstaste.
-	● Subtraktionstaste
SHIFT ^	● Wird benutzt, um Zahlen zu potenzieren. ● Wird benutzt, um das Exponenten-Anzeigesystem für numerische Daten in USING-Anweisungen zu spezifizieren.
SHIFT < SHIFT >	● Werden benutzt, wenn logische Operatoren in IF-SHIFT Sätzen eingegeben werden sollen.
DEF	● Wenn eines der folgenden 18 Zeichen (A,S,D,F,G,H,J,K,L,=,Z,X,C,V,B,N,M,SPaCe) nach Betätigen der DEF-Taste gedrückt wird, wird das Programm von der Programmzeile an gestartet, die das gleiche Etikett hat wie der Tastencode.

A ~ Z

- Buchstabentasten. Diese Tasten sind Ihnen wahrscheinlich von einer gewöhnlichen Schreibmaschine vertraut.
  Auf einfachen Tastendruck erscheinen Großbuchstaben in der Anzeige. In den Kleinschriftmodus schalten Sie um, indem Sie **SML** drücken.

SPC

- Wird beim Eingeben von Zeichen oder Programmen benutzt, um einen Leerschritt zu produzieren.

=

- In Zuweisungsanweisungen wird diese Taste benutzt, um der links vom "=" stehenden Variablen die rechts stehenden Inhalte (Zahlen oder Zeichen) zuzuweisen.
- Wird benutzt, wenn in IF-Sätzen logische Operatoren eingegeben werden.

SHIFT !
"
#
$
%
&
SHIFT @

- Zum Ansprechen der jeweiligen Symbole.
- ": • Zum Ansprechen und Löschen von Zeichen.
  • Zum Bestimmen von Etiketten.
  #: • In der USING-Anweisung wird hiermit die Anweisung für numerische Inhalte gegeben.
  $: • Wird zur Zuweisung von Zeichenfolgen-Variablen benutzt.
  &: • In der USING-Anweisung wird hiermit die Anweisung für Buchstabeninhalte gegeben.
  • Weist hexadezimale Zahlen aus.
  @: • Wird für den Reservespeicher benutzt, wenn die Reservetaste als Programmtaste eingesetzt wird.
  Beispiel: GPTO 1Ø0
  !% • Wird als Zeichenfolge innerhalb " " ver wendet.

SHIFT ?

- Zur Eingabe von CLOAD?

:

- Zur Trennung von zwei oder mehr Befehlen in einer Programmzeile.

,
- Bezeichnet eine Pause zwischen zwei Gleichungen, zwischen Variablen oder Kommentaren.

;
- Zur Ausführung von Multi-Display (Anzeige von zwei oder mehr Werten/Inhalten zur gleichen Zeit).
- Schafft eine Pause zwischen Instruktion und Variable.

(**)**
- Zur Eingabe von Klammern.

►
- Bewegt den Cursor nach rechts (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Dauerfunktion)
- Zur Ausführung von Playback-Anweisungen.
- Ruft den Cursor, wenn er bei der Anzeige von Inhalten nicht angezeigt ist.
- Löscht bei manueller Rechnung Fehlermeldungen.

◄
- Bewegt den Cursor nach links (auf einmaligen Tastendruck Bewegung um eine Stelle; wird die Taste festgehalten, Dauerfunktion)
- Sonst wie unter der **►**-Taste beschrieben.

INS
- Fügt einen Leerschritt ein (das Zeichen ⎵ erscheint dabei) mit einer 1-Schritt-Kapazität zwischen der durch den Cursor angegebenen Adresse (N) und der vorhergehenden Adresse (N-1).

DEL
- Löscht den Inhalt der Stelle, auf der Cursor steht.

SHIFT **π**
- Zur Eingabe von Pi (π).

SHIFT **√**
- Zur Eingabe der Quadratwurzel.

ENTER

- Zur Eingabe einer Programmzeile in den Computer.
- Wird beim Schreiben von Programmen benötigt.
- Bedingt manuelle Rechnung oder direkte Ausführung eines Befehls durch den Computer.
- Zum Neustart eines Programms, das zeitweise durch ein INPUT- oder PRINT-Kommando unterbrochen wurde.

P←→NP

SHIFT **ENTER**

- Stellt den Druck- oder nichtdruckenden Modus her, wenn ein zusätzlicher Drucker angeschlossen ist.

SML

- Zum Anwählen und Aufheben des Kleinschriftmodus. (Schaltet die Anzeige SML an bzw, aus.)
- Die SML-Anzeige erscheint, wenn **SML** gedrückt wird. Wenn Sie nun die Tasten **A** **B** und **C** betätigen, werden a, b und c in der Anzeige ausgegeben. Durch erneutes Drücken von **SML** schalten Sie den Kleinschriftmodus wieder aus und es werden wieder Großbuchstaben in der Anzeige ausgegeben.

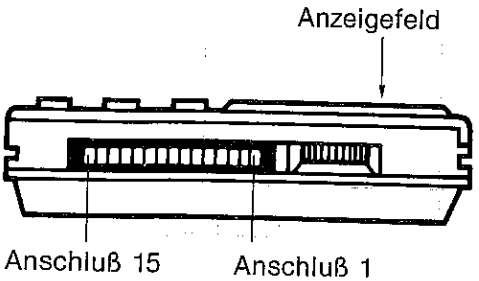
Die ↑ - und ↓ -Tasten haben folgende Funktionen, je nach dem angewählten Modus und Status des Computers:

Modus	Status	↓	↑
RUN	Programmdurchführung		
	Programm vorübergehend unterbrochen	Um die nächste Zeile durchführen zu lassen	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen
	PRINT, LINE, PRESET, PSET, GPRINT-Anweisung wird ausgeführt		
	PRINT-Anweisung wird momentan ausgeführt		
	Unterbrechung		
	INPUT-Anweisung wird momentan ausgeführt	keine Funktion	
	Fehlermeldung während der Programmausführung		Um fehlerhafte Zeile zur Anzeige zu bringen
	TRON-Status	Zur Fehlersuche	Diese Taste festhalten, um bereits abgearbeitete Zeilen zur Anzeige zu bringen.
PRO	(Wenn der Modus von RUN nach PRO geändert wird und die Programmzeile nicht angezeit wird)		
	Progdramm vorübergehend unterbrochen	Um die unterbrochene Zeile anzuzeigen	Wie links
	Fehler	Um die fehlerhafte Zeile anzuzeigen	Wie links
	Anderer Zustand	Um die erste Zeile anzuzeigen	Um die vorherige
	(Wenn die Progarmmzeile angezeigt wird)		
		Um die nächste Programmzeile anzuzeigen	Um die vorherige Programmzeile anzuzeigen
RESERVE			

- In der anzeige ist die ENTER Taste das gleiche wie ein Leerschritt.
- Wenn etwa 11 Minuten lang keine Taste betätigt wird, schaltet sich der Computer automatisch ab (automatische Abschaltung).

ANHANG F SYMBOLE DES E/A-ANSCHLUSSES

Der PC-1360 ist mit einem 15poligen Anschluß für ein E/A-Terminal ausgestattet. Die Anschlüsse und ihre Symbole werden anschließend beschrieben.



PC-1360
Ansicht von der rechten Seite

Benutzte Anschlüsse

Anschluß	Bezeichnung	Symbol	E/A	Funktion
1	Masseanschluß	FG	—	Masseanschluß für Wartung
2	Datenübertragung	SD	Aus	GS Ausgabesignal
3	Datenempfang	RD	Ein	GS Sendesignal
4	Sendeaufforderung	RS	Aus	HIGH: Sendet Trägersignal
5	Übertragungsbereit	CS	Ein	HIGH: Übertragung
7	Signalerde	SG	—	Bezugsspannung Ø für alle Signale
8	Trägersignal erh.	CD	Ein	HIGH: Trägersignal erhalten
10		VC		Energieversorung
11	Empfangsbereit	RR	Aus	HIGH: Empfang möglich
12	Periph. Quittung	PAK	Ein	Siehe Anmerkung 3
13		VC	—	Energieversorgung
14	Datenterm. bereit	ER	Aus	HIGH: Daten-Terminal bereit
15	Periph. Quittung	PRQ	Aus	Siehe Anmerkung 3

Beachte:
1. HIGH: VC-Spannungspegel; LOW: SG-Spannungspegel.
2. Der Computer benutzt CMOS-Komponenten. Eine Anwendung von den zulässigen Bereich, d.h. den Spannungspegel zwischen SG und VC überschreitenden Spannungen, könnte die Beschädigung des Computers zur Folge haben.
3. Das Signal bestätigt, ob das angeschlossene Endgerät der CE-140P-Drucker ist oder nicht. Dieses Signal wird nicht für die Kommunikation mit anderen Endgeräten verwendet.

ANHANG G TECHNISCHE DATEN

Modell: PC-1360 Taschencomputer
Prozessor: 8 Bit CMOS CPU
Programmiersprache: BASIC
System-ROM: 136 K Bytes
Speicherkapazität: RAM:

	Systemintern	960 Bytes	
In RAM-Karte gespeichert (8kB)	Systembereich	1282 Bytes	Kann durch Verwendung der beiden Schlitze auf 64kB erweitert werden
	Verwender		
	Fix. Speicherbereich (A ~ Z, A$ ~ Z$)	208 Bytes	
	Prog./Daten-Bereich	6558 Bytes	
	Reservierter Bereich	144 Bytes	

Stack: Subroutine: 10 Stacks Funktion: 16 stacks
FOR-NEXT: 5 Stacks Daten: 8 stacks

Operatoren: Addition, Subtraktion, Multiplikation, Division, trigonometrische Funktionen und Arkusfunktionen, logarithmische und Exponentialfunktionen, Winkelumkehrfunktionen, Quadrat und Quadratwurzel, Vorzeichen, Absolut, Integral, Verhältnis-Operatoren, logische Operatoren usw.

Rechengenauigkeit: 10 Stellen (Mantisse) + 2 Stellen (Exponent)

Aufbereitungsmöglichkeiten:
Cursor rechts und links, Zeile auf- und abwärts, Zeicheneinfügung, Zeichenlöschung

Speicherschutz: CMOS, batteriegestützt

Charakteristika des seriellen E/A-Anschlusses:

Standards:	Start-Stopp-Übertragunssytem (asynchron). Nur Halb-Duplex.
Baud-Raten:	300, 600, 1200 Baud
Datenbits:	7 oder 8 Bits
Paritätsbits:	Gerade, ungerade oder keine Parität

Stoppbit:	1 oder 2 Bits
Anschlüsse:	15poliger Anschluß (für externe Geräte)
Ausgabesignal:	C-MOS-Niveau (4—6 Volt)
Interface-Signale:	Eingaben: RS, CS, CD Ausgaben: SD, RS, RR, ER Andere: SG, FG, VC
Drucker-Interface:	11poliger Anschluß
Grafik-Interface:	CE-140P Kommandos (CIRCLE, PAINT usw.)
Einbauplätze für RAM-Karten:	jede Karte 2KB, 4KB, 8KB, 16KB oder 32 KB
Anzeige:	4-Zeilen-, 24-Stellen-Flüssigkristallanzeige mit 5 × 7 Punkt-Zeichen oder Grafik mit 150 × 32 Punkten.
Tastatur:	62 Tasten. Alphabetisch, numerisch, Sonderzeichen und Funktionen. Numerische Schablone Vom Benutzer wählbare Tasten.
Stromversorgung:	6.0 V DC: Lithium-Zellen. Typ: CR-2032 × 2
Stromverbrauch:	6.0 V DC @ 0.03W Unter normalen Umständen 120 Stunden ununterbrochener Betrieb (bei angenommenen 10 Minuten Bedienung und 50 Minuten Anzeigezeit per Stunde bei einer Temperatur von 20°C) möglich. Die Betriebszeit kann je nach Verwendung und Batterietype leicht schwanken.
Betriebstemperatur:	0°C — 40°C
Abmessungen:	182 (B) × 72 (T) × 16 (H) mm
Gewicht:	Ca. 220 g (mit Batterien und einer RAM-Karte)
Zubehör:	Deckel, zwei Lithium-Zellen (eingebaut), eine Tastaturschablone, eine 8KB RAM-Karte (CE-212M), und ein Handbuch.

Zusatzgeräte:

RAM-Einsteckkarten 2kB (CE-210M)
4kB (CE-211M)
8kB (CE-212M)
16kB (CE-2H16M)
32kB (CE-2H32M)

Kassettenrecorder (CE-152)
Drucker/Kassetten-Interface (CE-126P).
Drucker (CE-140P, CE-515P, CE-516P) usw.

Kopiermöglichkeit:

RAM-Karten können vom Einbauplatz 1 zum Einbauplatz 2 kopiert werden.

ANHANG H BENUTZUNG VON PROGRAMMEN AUF PC-1360, DIE FÜR ANDERE SHARP MODELLE ENTWICKELT WURDEN

Dieses Kapitel enthält Information über die Programmkompatibilität für die folgenden Modelle:

PC-1260, PC-1261
PC-1350
PC-1401, PC-1402
PC-1450, PC-1460

PC-1260, PC-1261

1. Diese Modelle haben nicht das 4-Zeilen-Anzeigefeld des PC-1360, Benutzen Sie das CLS-Kommando, um die Anzeige nach jeder Eingabe von ein oder zwei Zeilen zu löschen.

2. Jeder Zeichencode 96 (&6Ø), der einen Leerraum angibt, muß in den Code 32 (&2Ø) umgewandelt werden, da der Code 96 ein einziges Anführungszeichen auf dem PC-1360 angibt. Diese Codes werden normalerweise in den CHR$-Kommandos verwendet.

PC-1350

1. Programme, die mit dem PC-1350 geschrieben wurden, können grundsätzlich mit dem PC-1360 verwendet werden. Wenn der PC1360 ein vom Band ein Programm des PC-1350 einliest, wird dieses automatisch umgewandelt. Abhängig vom Programm kann es aber zu einem Spei herüberlauf oder zum Verlust von Variablen kommen, weil das Programm zu groß ist. Ebenfalls abhängig vom Programm kann es zu einem Fehler kommen, wenn eine Programmzeile mehr als 80 Bytes enthält. UmM PC-1360 Programme mit dem PC-1350 zu verwenden, können Sie es mit dem folgenden Befehl auf Band speichern:

CSAVE@ ["Dateiname"][,"PASSWORD"] ENTER

Beachte: Mit dem obigen Befehl ist es nicht möglich, reservierte Inhalte zu speichern, da diese bei beiden Computern verschieden sind.

PC-1401, PC-1402

1. Die , -Taste ist keine definierte Taste in diesem Computer. Für Programme, die die , -Taste vorgeben geben Sie eine andere Taste vor. Zum Beispiel:

   100 ",": (Wechseln Sie auf) 100 "=":

2. Der PC-1401 hat mehr Funktionskommandos als der PC-1360. Sie erhalten Fehlermeldungen, wenn das Programm auf dem PC-1360 diese Kommandos enthält, die nicht identifiziert werden können. wenn das Programm in den PC-1360 eingelesen wird, werden die Kommandos mit dem Symbol "~" anstelle des Kommandos angezeigt.

PC-1450, PC-1460

Diese Modelle haben nicht das 4-Zeilen-Anzeigefeld des PC-1360. Benutzen Sie das CLS-Kommando, um die Anzeige nach jeder Eingabe von ein oder zwei Zeilen zu löschen.

Programmbeispiele

Nachdem Sie jetzt die Beschreibung der verschiedenen Funktionen in den vorangegangenen Kapiteln gelesen habaen, haben Sie bereits eine Reihe von Programmbefehlen kennengelernt. Um allerdings Ihr Wissen zu verbessern und praktisch anwenden zu können, ist es unumgänglich, daß Sie Ihre eigenen sowie die in dieser Bedienungsanleitung befindlichen BASIC-Programme schreiben und ablaufen lassen.

Wie Ihnen nur das echte Steuern eines Wagens oder das Schwingen eines Tennisschlägers helfen wird, Ihre Fähigkeiten auf diesen Gebieten zu verbessern, so erreichen Sie echte "Programmierfähigkeit" nur durch das Schreiben von so vielen Programmen wie möglich, ungeachtet Ihers derzeitigen Niveaus oder Fähigkeiten.

Es ist auch sehr wichtig, daß Sie Programme, die von anderen entwickelt wurden, studieren. In diesem kapitel werden Sie mit einigen Programmbeispielen, die verschiedene BASIC-Befehle verwenden, bekannt gemacht.

Zum besseren Verständnis der Programmbeispiele in diesem Kapitel, werden nachfolgend die in diesen Beispielen verwendeten Begriffe erklärt.

1. PROGRAMMLISTE
   Alle in den Programmbeispielen enthaltenen Programmlisten können mittels der Hartkopien-Ausgabe über den CE-126P oder den CE-140P in tatsächlicher Größe oder in 55% reduzierter Größe ausgegeben werden.

2. PROGRAMM-KAPAZITÄT
   Am Ende jeder Programmliste wird die Programm-Kapazität in Form der Byte-Anzahl angegeben.

3. AUSDRUCK
   Bei einem Programm, welches einen Ausdruck erfordert, erfolgt die Ausgabe unter Verwendung des CE-126P oder CE-140P Druckers in tatsächlicher Größe oder 55% reduziert.
   Ein Ausdruck, der um einen anderen Wert als 55% reduziert wurde, wird mit dem entsprechenden Reduktionswert markiert.

4. SPEICHERINHALT

In der in jedem Programmbeispiel enthaltenen Tabelle des Speicherinhalts, werden Variablen mit einer vorbestimmten Verwendung durch Ihre spezielle Verwendung gekennzeichnet und diese ohne vorbestimmte Verwendung (z.B. diejenigen, die im Arbeitsspeicher gespeichert werden um die Zwischenergebnisse von Rechnungen zu speichern, etc.) werden durch die Prüfmarke "√" gekennzeichnet.

5. VERWENDUNG DES CE-515P (oder des CE-516P)

Im Inhaltsverzeichnis dieses Kapitels bedeutet ein einem Programmnamen voranstehendes "P", daß der CE-140P, CE-515P oder CE-516P bei der Ausführung des Programmes als ein Peripheriegerät benutzt werden kann. Bei Verwendung des CE-515P oder CE-516P sollten jedoch die folgenden Punkte beachtet werden:

1. DIP-Schalter Stellung
   - Mit dem CE-515P..... Stellen Sie alle DIP-Stiftschalter auf OFF.
   - Mit dem CE-516P..... Stellen Sie die Stiftschalter 1 bis 5 auf OFF und den Stiftschalter 6 auf ON.

2. Zu verwendendes Papier
   Verwenden Sie in jedem Fall Rollenpapier.

Beachte: Wenn Sie den CE-515P oder den CE-516P verwenden, unterscheiden sich die Hartkopien teilweise von den des CE-140P in folgenden Punkten:

(1) Farbe
CE-140P: purpurn → CE-515P (oder CE-156P)

(2) Zeichen
CE-140P:a → CE-515P: α
(CE-516P druckt das selbe Zeichen als der CE-140P).

Die SHARP CORPORATION und/oder angegliederte firmen übernehmen keinerlei Verantwortung oder Haftung für eventuelle Schäden oder finanzielle Verluste, die durch Anwendung der in dieser Bedienungsanleitung enthaltenen Programmbeispiele auftreten mögen. Bei Anwendung der Programmbeispiele sollten Sie im Sinn behalten, daß diese Programme nicht völlig dem Zweck, für den Sie sie einsetzen, entsprchen mögen oder daß manche Programme nicht so genau sind, wie Sie sich das wünschen. Prüfen sie daher die Daten der verwendeten Programmbeispiele und stellen Sie fest, ob sie Ihren Anforderungen entsprechen. Ist dies nicht der Fall, modifizieren Sie sie den ihnen zugedachten Zweck entsprechend.

INHALT

(Programmname)

1. VERGLEICHENDE BAND-GARFIK 342
2. MODIFIZIERTER BEWEGLICHER MITTELWERT 350
3. ÜBERTRAGUNG EINER PROGRAMMDATEI 353
4. RUNDE GRAFIK 358
5. SKISPRUNG 362

Programmname: VERGLEICHENDE BAND-GRAFIK

Farb-Matrixdrucker
CE-140P erforderlich

■ ÜBERBLICK

Diese Grafik wird zum Vergleich des Verhältnisses einzelner Importposten zueinander in Prozent verwendet.

■ ANWEISUNGEN

1. Das Programm wird mit den Tasten **DEF** und **A** gestartet.
2. Geben sie die anzahl der Bänder ein (1-25).
3. Entsprechend der anzeige, geben Sie wie folgt ein: Name der Bänder (nicht mehr als sieben Zeichen), anzahl der Posten (zwischen 2-12) und die Bezeichnung der Artikel (nicht mehr als sieben Zeichen).
4. Punkt 3 folgend, geben sie die Daten jedes Postens ein. Es ist möglich, bis zu 8-stellige Zahlen einzugeben.
5. Geben Sie den Namen der Grafik ein (nicht mehr als acht Zeichen). Das Programm wird nach dem Ausdruck der Datentabelle und der vergleichenden Grafikbänder beendet.

■ ANMERKUNGEN

Durch Ändern der Daten in Zeile 580 können Sie die Farben sowie das Muster der Grafik ändern.

CL	0	1	2	3	4	5	6
FARBE	SCHWARZ	PURPURN	ROT	MAGENTA	GRÜN	ZYAN	GELB

PT	1	2	3
MUSTER	HORIZONTAL	VRTIKAL	KREUZENDE STREIFEN (SCHRAFFIERUNG)

Wenn Sie 42 angeben, werden grüne vertikale Linien ausgedruckt.

■ BEISPIEL

Mittels der nachfolgenden Daten wird eine vergleichende Bandgrafik erstellt.

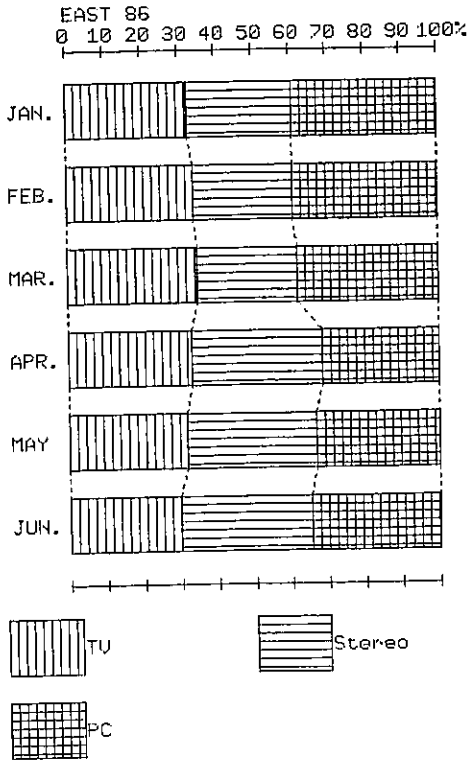
Anzahl der Bänder: 6
Anzahl der Posten: 4

Bezeichnung der Grafik: OSTEN 86

Posten / Monat	TV	Stereo	PC
JAN.	30	26	36
FEB.	28	22	32
MAR.	26	20	28
APR.	34	36	32
MAY	29	32	30
JUN.	25	30	29

■ AUFBAU

	TV	Stereo	PC
JAN.	30	26	36
FEB.	28	22	32
MAR.	26	20	28
APR.	34	36	32
MAY	29	32	30
JUN.	25	30	29



■ TASTENFOLGE

1.) DEF A

```
NO. OF BELT = _
```

2.) 6. ENTER

```
NAME OF BELT(1) =
?
```

3.) JAN. ENTER

```
NAME OF BELT(2) =
?
```

Eingabe wie oben

6.) JUN. ENTER

```
NO. OF ITEM = _
```

7.) 3 ENTER

```
NAME OF ITEM(1) =
?
```

8.) TV ENTER

```
NAME OF ITEM(?) =
?
```

Eingabe wie oben

10.) PC ENTER

```
JAN.
TV
```

```
DATA = _
```

11.) 30 ENTER

```
JAN.
STEREO
```

```
DATA = _
```

Eingabe wie oben

26.) 29 **ENTER**

```
GRAPH TITLE=
?
```

27.) EAST 86 **ENTER**

```
>
```

■ PROGRAMMLISTE

```
 10:"A": WAIT 0: CLEAR : CLS : USING
 20:INPUT "NO. OF BELT=";YK
 30:IF YK<1 OR YK>25 BEEP 2: GOTO 10
 40:DIM Y$(YK),T$(0)*18,YT(YK)
 50:FOR I=1 TO YK
 60:CLS : PRINT "NAME OF BELT(";STR$ I
    ;")=": INPUT Y$(I)
 70:IF LEN Y$(I)>7 BEEP 2: GOTO 60
 80:NEXT I
 90:CLS : INPUT "NO. OF ITEM=";XK
100:IF XK<2 OR XK>12 BEEP 2: GOTO 90
110:DIM X$(XK),DD(XK,YK),XT(XK),CX(XK-1
    ),HP(XK)
120:FOR I=1 TO XK
130:CLS : PRINT "NAME OF ITEM(";STR$ I
    ;")=": INPUT X$(I)
140:IF LEN X$(I)>7 BEEP 2: GOTO 130
150:NEXT I
160:FOR Y=1 TO YK: FOR X=1 TO XK
170:CLS : PRINT Y$(Y): PRINT X$(X)
180:D=0: INPUT "DATA=";D
190:D= INT D: IF LEN STR$ D>8 BEEP 2:
    GOTO 170
200:DD(X,Y)=D: NEXT X: NEXT Y
210:CLS :T$(0)="": PRINT "GRAPH TITLE="
    : INPUT T$(0)
220:IF LEN T$(0)>8 BEEP 2: GOTO 210
230:CLS : GOSUB 1170
240:S1= INT (XK/5)+1:X=0:A=96
250:FOR O=1 TO S1
260:S2=4: IF O=S1 LET S2=XK-5*O+5
270:GLCURSOR (0,-20): SORGN
280:X2=A*(S2+1)
290:LLINE -(X2,0),L1
300:FOR I=0 TO S2+1
310:LLINE (X2-I*A,0)-(X2-I*A,-24),L1:
    NEXT I
320:FOR I=0 TO S2
330:GLCURSOR (A*I+8,-18): LPRINT "P";X$
    (X+I): NEXT I
340:FOR Y=1 TO YK: GLCURSOR (0,-24):
    SORGN
350:RLINE -(X2,0),L1
360:FOR I=0 TO S2+1
370:LLINE (X2-I*A,0)-(X2-I*A,-24),L1:
    NEXT I
380:FOR I=0 TO S2
390:IF X+I=0 GLCURSOR (A*I+8,-18):
    LPRINT "P";Y$(Y): GOTO 410
400:GLCURSOR (A*I-5,-18): LPRINT CHR$ 2
    7;"c11": LPRINT "P"; USING "########
    ##";DD(X+I,Y)
410:LPRINT CHR$ 27;"c12": NEXT I: NEXT
    Y
420:LLINE (X2,-24)-(0,-24),L1: SORGN
430:X=X+4: NEXT O
440:LTEXT
450:ZX=350:ZY=500
460:MS=0: FOR K=1 TO YK:L= LEN Y$(K):
    IF L>MS LET MS=L
470:NEXT K: IF MS=1 LET MS=2
480:PY=-60:X=MS*12+6
490:IF (ZX+MS*12+51)>480 LET ZX=480-MS*
    12-51
500:IF (ZX+X+36)>480 LET ZX=480-X
510:PX= INT ((480-ZX)/2): IF PX<X LET P
    X=X
520:XH= INT (ZX/10):PA=XH/10
530:KH= INT (ZY/(YK*2+(YK+1))):OB=KH*2
540:IF OB>75 LET OB=75:KH=38
550:RESTORE 560: FOR I=1 TO XK: READ H
    P(I): NEXT I
560:DATA 11,22,33,02,12,23,31,01,13,21,
    32,03
570:FOR K=1 TO XK:XT(K)=0: FOR I=1 TO Y
    K:XT(K)=XT(K)+DD(K,I): NEXT I: NEXT
    K
580:FOR K=1 TO YK:YT(K)=0: FOR I=1 TO X
    K:YT(K)=YT(K)+DD(I,K): NEXT I: NEXT
    K
590:GOSUB 1170: USING : GLCURSOR (PX,-
    30): CSIZE 2: LPRINT "P";T$(0)
600:GLCURSOR (PX,PY): SORGN
610:YU=5:YD=-5:XM=-XH:L=12: IF ZX<300
    LET L=6
620:CSIZE 2
630:FOR K=0 TO 10:XM=XM+XH:AX=.4*L
640:IF K>=1 LET AX=.9*L
650:IF K=10 LET AX=1.3*L
660:GLCURSOR (XM-AX,YU*2)
670:LPRINT "P"; STR$ (K*10): IF K=10
    LPRINT "P%"
680:LLINE (XM,YU)-(XM,YD),L1: NEXT K
```

```
690:LLINE (XM,YD+YU)-(0,0),L1: CSIZE 2
700:SY=OB-4
710:FOR KA=1 TO YK:SX=0:SY=SY-OB-KH
720:MX=-(MS*12+6):MY=SY-KH-6
730:GLCURSOR (MX,MY): LPRINT "P";Y$(KA)
740:TT=0:PN=0
750:FOR K=1 TO XK:PN=PN+1:TT=TT+DD(K,KA
    )
760:IF YT(KA)=0 LET EX=0: GOTO 780
770:PC=TT/YT(KA)*100:EX= INT (PC*PA)
780:IF EX=SX AND K<>1 THEN 820
790:LLINE (SX,SY)-(EX,SY),L1
800:LLINE -(EX,SY-OB),L1
810:LLINE -(SX,SY-OB),L1
820:IF K=1 RLINE -(0,OB),L1
830:IF KA=1 THEN 910
840:L1=2
850:IF K=XK LLINE (EX,SY)-(EX,SY+KH),L1
860:IF K=XK THEN 900
870:IF K=1 LLINE (SX,SY)-(SX,SY+KH),L1
880:LLINE (CX(K),SY+KH)-(EX,SY),L1
890:LLINE -(EX,SY),L1
900:L1=0
910:IF EX<>SX GOSUB 1190
920:SX=EX: IF K<XK LET CX(K)=EX
930:NEXT K: NEXT KA
940:SX=0:SY=SY-OB-KH-5
950:GLCURSOR (SX,SY): SORGN
960:YU=5:YD=-5:XM=-XH:SW=1
970:FOR K=0 TO 10:XM=XM+XH
980:IF SW<>1 LLINE (XM,YD)-(XM,YU),L1:
    GOTO 1000
990:LLINE (XM,YU)-(XM,YD),L1
 1000:SW=-SW: NEXT K
 1010:LLINE (XM,YD+YU)-(0,0),L1
 1020:GLCURSOR (0,-KH-5): SORGN
 1030:SY=0:S1=1: FOR PN=1 TO XK
 1040:IF INT S1<0 LET SX=XH*5:EX=SX+XH*
      2: GOTO 1060
 1050:SX=-60:EX=XH*2-60
 1060:LLINE (SX,SY)-(EX,SY),L1
 1070:LLINE -(EX,SY-OB),L1
 1080:LLINE -(SX,SY-OB),L1
 1090:LLINE -(SX,SY),L1
 1100:GOSUB 1190:X=EX+3:Y=SY-OB/2-6
 1110:GLCURSOR (X,Y): LPRINT "P";X$(PN)
 1120:IF S1<0 LET SY=SY-OB-KH
 1130:S1=-S1: NEXT PN
 1140:SORGN
 1150:GLCURSOR (0,-100)
 1160:LPRINT CHR$ 27+"@": END
 1170:LPRINT : LPRINT : CSIZE 2: LPRINT
      : LTEXT : LPRINT : GRAPH
 1180:COLOR 0:L1=0: SORGN : RETURN
 1190:CL= INT (HP(PN)/10):PT=HP(PN)-CL*
      10:EY=SY-OB
 1200:COLOR CL,7
 1210:ON PT GOSUB 1240,1290,1340
 1220:COLOR 0
 1230:RETURN
 1240:HH=8:SW=1
 1250:FOR X=SX+HH TO EX STEP HH: IF X>E
      X THEN 1280
 1260:IF SW<>1 LLINE (X,EY)-(X,SY),L1:
      GOTO 1280
 1270:LLINE (X,SY)-(X,EY),L1
 1280:SW=-SW: NEXT X: RETURN
 1290:HH=8:SW=1
 1300:FOR Y=SY-HH TO EY STEP -HH: IF Y<
      EY THEN 1330
 1310:IF SW<>1 LLINE (EX,Y)-(SX,Y),L1:
      GOTO 1330
 1320:LLINE (SX,Y)-(EX,Y),L1
 1330:SW=-SW: NEXT Y: RETURN
 1340:GOSUB 1240: GOSUB 1290: RETURN
```

3207 bytes

■ SPEICHERINHALTE

Variablen	Inhalte	Variablen	Inhalte
A	√	PX	Nullpunkt von X
D	Eingangsdaten	PY	Nullpunkt von Y
I	Schleifenzähler	S1	√
K	Schleifenzähler	S2	√
L	√	SW	Flag
M	√	SX	Grafikdruck
O	Schleifenzähler	SY	Grafikdruck
W	√	TT	√
X	√	X2	√
Y	√	XH	√
AX	Meßdruck	XK	Anzahl der Posten
CL	Farben	XM	Meßdruck
EX	Grafikdruck	YD	Meßdruck
EY	Grafikdruck	YK	Anzahl der Bänder
HH	Breite der Schraffierung	YU	Meßdruck
KA	Schleifenzähler	ZX	Länge der Bänder
KH	Breite der Bänder	ZY	√
LI	√	CX (XK-1)	Verbundene Daten
MS	√	HP (XK)	Schraffierungs-Daten
MX	Druckposten	XT (XK)	Summe jedes Postens
MY	Druckposten	YT (XK)	Summe jedes Bandes
OB	Breite des Bandes	DD (XK, YK)	Daten
PA	√	T$ (O)	Titel
PC	Druckband	X$ (XK)	Name des Posten
PN	Schleifenzähler	Y$ (YK)	Name des Bandes
PT	Für das Schraffieren		

■ ZEILENPLAN

Zeilennummer	Inhalt
10	Initialisierung
20 ~ 40	Anzahl der Band-Eingaben
50 ~ 80	Bezeichnung der Band-Eingaben
90 ~ 110	Anzahl der Posten-Eingaben
120 ~ 150	Bezeichnung der Posten-Eingaben
160 ~ 200	Dateneingabe
210 ~ 220	Eingabe d. Grafik-Titel
230 ~ 430	Grafikaten-Ausdruck
440 ~ 580	Berechnung
590	Ausdruck d. Grafik-Titel
600 ~ 690	Oberer Meßdruck
700 ~ 730	Grafikdruck
940 ~ 1000	Unterer Meßdruck
1010 ~ 1130	Kommentar-Ausdruck
1140 ~ 1150	Ende
1160	Drucker Initial. Sub.
1170 ~ 1230	Schraff. Muster Verarbeitung
1240 ~ 1280	Horizont. Schraffierung Sub.
1290 ~ 1330	Vertik. Schraffierung Sub.
1340	Horizont/Vert. Schraffierung Sub.

Programmname: MODIFIZIERTER BEWEGLICHER MITTELWERT

Dafür das
Thermaldrucker/Kassetten-Interface
CE-126P benötigt.

■ UBERBLICK

Selbst schwer erkennbare Veränderungen können leicht erfaßt werden! Der modifizierte bewegliche Mittelwert wird überall dort eingesetzt, wo aus einer festen Datenfolge der Mittelwert an jedem Punkt ermittelt werden soll und wenn Sie Datentrends erkennen wollen.
Mit dieser Methode können Langzeittrends in solch flüchtigen Indikatoren wie Verkaufszahlen, Aktienpreise und Wechselkurse deutlich gemacht werden.

■ ANWEISUNGEN

1. **DEF** **A** startet das Programm. Geben Sie dann die Anzahl der Werte und dann die einzelnen Werte ein. Wird ein Wert eingegeben, so werden die Eingabedaten vom Drucker ausgegeben und darüber hinaus werden die Eingabedaten und die Mittelwerte von n + 1. an auf dem Drucker ausgegeben.

2. Um das Programm zu beenden, drücken Sie nur die **ENTER**-Taste bei der Abfrage der Dateneingabe.

■ INHALT

Es gibt zwei verschiedene Versionen, je nachdem, ob die Zahl der Eingaben (n) gerade oder ungerade ist.

1. n ist eine ungerade Zahl:

$$\bar{X}_1 = \sum_{i=1}^{n} Xi/n$$

$$\bar{X}_1 = \sum_{i=2}^{n+1} Xi/n$$

2. *n ist eine gerade Zahl:*

$$\bar{X}_1 = \left(\frac{X_1}{2} + \frac{X_{n+1}}{2} + \sum_{i=2}^{n} Xi\right)/n$$

$$\bar{X}_2 = (\frac{X_2}{2} + \frac{X_{n+2}}{2} + \sum_{i=3}^{n+1} Xi)/n$$

⋮

■ BEISPIEL

Geben Sie die folgenden Daten ein, um den modifizierten beweglichen Mittelwert zu ermitteln. Setzen Sie dazu die Zahl der Eingaben auf 4.

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}
82	15	8	53	Ø	41	22	3	18	75	64	39	46

■ GEDRUCKTE AUSGABE

```
* x(1)=82.
* x(2)=15.
* x(3)=8.
* x(4)=53.
* x(5)=0.
    Average=29.25
* x(6)=41.
    Average=22.25
* x(7)=22.
    Average=27.25
* x(8)=3.
    Average=22.75
* x(9)=18.
    Average=18.75
* x(10)=75.
    Average=25.25
* x(11)=64.
    Average=34.75
* x(12)=39.
    Average=44.5
* x(13)=46.
    Average=52.5
```

■ TASTENFOLGE

1) **DEF** **A** [Programmstart]

```
Number of items = _
```

2) 4 **ENTER** [Eingabe der Anzahl der Werte]

```
Number of items = 4
X (1) = ?
```

■ PROGRAMMLISTE

```
10:"A": CLEAR : CLS :
   WAIT 0: LPRINT "":
   USING
20:N=1:E=0:C=1
30:INPUT "Number of ite
   ms = ";A
40:DIM X(A-1)
50:IF A<> INT (A/2)*2
   THEN 120
60:FOR I=0 TO A-1:
   GOSUB 200: NEXT I
70:FOR I=0 TO A-1:
   GOSUB 220: CURSOR 5+
   LEN STR$ N,C: INPUT
   D: GOTO 90
80:GOTO 250
90:F=X(I): GOSUB 210
100:LPRINT "     Average=
   ";(E-.5*(D+F))/A
110:E=E-F: NEXT I: GOTO
   70
120:FOR I=0 TO A-2:
   GOSUB 200: NEXT I
130:B=A-1: GOSUB 220:
   CURSOR 5+ LEN STR$ N
   ,C: INPUT D
140:GOSUB 230:X(B)=D
150:GOSUB 240
160:FOR I=0 TO B: GOSUB
   220: CURSOR 5+ LEN
   STR$ N,C: INPUT D:
   GOTO 180
170:GOTO 250
180:E=E-X(I): GOSUB 210
190:GOSUB 240: NEXT I:
   GOTO 160
200:GOSUB 220: CURSOR 5+
   LEN STR$ N,C: INPUT
   D
210:GOSUB 230:X(I)=D:
   RETURN
220:PRINT : PRINT "x("+
   STR$ N+")=": RETURN
230:LPRINT "* x("+ STR$
   N+")=";D:E=E+D:N=N+1
   :C=C+(C<>3): RETURN
240:LPRINT "     Average
   =";E/A: RETURN
250:LPRINT : LPRINT :
   CLS : END
```

575 bytes

3) 82 **ENTER** [Eingabe von X(1)]

```
Number of items = 4
X(1) = 82
X(2) = ?
```

Gleiche Eingabe wie hier oben

4) 46 **ENTER** [Eingabe vonX(13)]

```
X(11) = 64
X(12) = 39
X(13) = 46
X(14) = ?
```

5) **ENTER**]Programm beendet]

```
>
```

■ SPEICHERINHALTE

A	Anzahl der Posten
B	√
C	√
D	Daten
E	Alle Daten
F	√
I	Schleifenzähler
N	√
X(A—1)	Daten

· Programmname: ÜBERTRAGEN EINER PROGRAMMDATEI

Dafür wird der
CE-130T benötigt

In der Vergangenheit waren nur Cassettenbänder (Mikrocassetten) als externe Speicher für Taschencomputer verfügbar. Nun kann aber eine Diskette in einem Personal-Computer mittels des CE-130T Pegelumwandlers für den PC-1360 Taschencomputer als externer Speicher dienen.
In diesem Beispiel wird für die Übertagung von Programmdateien zu und vom PC-1360 Taschencomputer stellvertretend für die Personal-Computer der PC-7000 verwendet.

■ VORBEREITUNG

- Zusätzlich zum PC-7000 Personal-Computer müssen Sie auch ein Kabel für die verbindung des CE-130T zum RS-232C Interface des PC-7000 zur Verfügung haben. Die folgende Aufstellung zeigt die Anforderungen an das Kabel und die Steckerbelegung.

CE-130T

Signalbezeichnung	Pin Nr.
—	1
SD	2
RD	3
RS	4
CS	5
DR	6
SG	7
CD	8
RR	11
ER	20

Vaterstecker
DB-25P oder gleichwertig

PC-7000

Pin Nr.	Signalbezeichnung
1	FG
2	SD
3	RD
4	RS
5	CS
6	DR
7	SG
8	CD
20	ER

Mutterstecker
DB-25S oder gleichwertig

.....bedeutet, daß die Verbindung dieser beiden Anschlußstifte nicht unbedingt notwendig ist.

- ÜBERTRAGEN EINER PROGRAMMDATEI (PC-1360 → PC-7000)
  hier wird Ihnen die Methode der Übertragung einer Programmdatei vom PC-1360 zum PC-7000 vorgestellt. Der PC-7000 wird die vom PC-1360 erhaltenen Daten auf einer 5-1/4-Zoll Diskette speichern. Der Inhalt der empfangenen Programmdatei kann auf dem Bildschirm des PC-7000 nachgeprüft werden.

■ BEDIENUNGSVORGANG

PC-1360
Programmeingabe

Geben sie das an den PC-7000 zu übertragende Programm ein.

Vorbereitung für Programmübertragung

Stelle auf PRO- od. RUN-Modus.
CLOSE **ENTER**
OPEN "1200, N, 8, 1, A, C, &1A"
ENTER
→ Der SAVE-Befehl ist nun durchführbar

Übertragung der Programmdatei

SAVE **ENTER**.
→ Der Computer überträgt die Programmdatei.

PC-7000
Programmeingabe

Geben sie Programm 1-1, ein (Siehe unten).

Empfang der Programmdatei

RUN ↵
→ Der Computer führt Programm 1-1. aus.
→ Der Computer wartet auf den Empfang d. Programmdatei.

Der computer empfängt die Programmdatei.
→ Der Inhalt der erhaltenen Programmdatei wird zeilenweise auf dem Bildschirm angezeigt.
Geben Sie den Namen der empfangenen Programmdatei ein.
→ Die Speicherung der Programmdatei auf Diskette ist nun abgeschlossen.

■ PROGRAMMLISTE 1-1

```
10 CLEAR:DIM A$(1000)
20 CLOSE:OPEN "COM1:1200,N,8,1" AS #1
30 Z$=INPUT$(1,1):IF Z$=CHR$(&H1A) THEN 100
40 LINE INPUT #1,A$
50 A$=Z$+A$
60 PRINT A$
70 A$(I)=A$
80 I=I+1
90 GOTO 30
100 INPUT "FILE NAME=";FI$
110 OPEN "O",#2,FI$
120 FOR J=0 TO I-1
130 PRINT #2,A$(J)
140 NEXT J
150 CLOSE
160 END
```

Für den PC-1360 ist eine Programmierung für die Übertragung nicht erforderlich.

■ SPEICHERINHALTE

I	Zähler
J	Schleifenzähler
A$	Für Eingabe einer Programmzeile
Z$	√
FI$	Dateiname
A$(1000)	√

■ ZEILENPLAN

Zeilennummer	Inhalt
10 ~ 20	Initialisierung
30 ~ 90	Emfang d. Programmes durch RS-232C.
100	Eingabe des Dateinamen.
110 ~ 160	Das empfangene Programm ist auf Diskette gespeichert.

- ÜBERTRAGEN EINER PROGRAMMDATEI (PC-7000 → PC-1360)
  Hier wird die methode der Übertragung einer Programmdatei von der Diskette des PC-7000 zum PC-1360 beschrieben. Der Inhalt der übertragenen Programmdatei kann auf dem Bildschirm des PC-7000 nachgeprüft werden.

■ OPERATING PROCEDURE

PC-7000
Programmeingabe

Geben Sie Programm 2-1. ein (Siehe unten).

Übertragung der Programmdatei

RUN [↵]
→ Der Computer führt Programm 2.1. aus. Geben Sie den Namen der zu übertragenden Datei ein.
→ Der Inhalt der übertragenen Programmdatei wird zeilenweise angezeigt.
→ Der Computer überträgt die Programmdatei.

PC-1360
Vorbereitungen für Empfang des Programmes.

Schalten Sie auf den PRO- oder RUN-Modus.
CLOSE **ENTER**
OPEN "1200,N,8,1,A,C,&1A"
ENTER
→ Der LOAD-Befehl ist nun durchführbar.

Empfang der Programmdatei

LOAD **ENTER**
→ Der Computer wartet auf den Empfang der Programmdatei.
Der Computer empfängt die Programmdatei.
→ Ende der Übertragung.

■ PROGRAMMLISTE 2-1

```
10 CLOSE:OPEN "COM1:1200,N,8,1" AS #1
20 INPUT "FILE NAME=";FI$
30 OPEN "I",#2,FI$
40 IF EOF(2)=-1 THEN PRINT #1,CHR$(&H1A);:CLOSE:END
50 LINE INPUT #2,A$
60 PRINT A$
70 PRINT #1,A$+CHR$(&HD);
80 GOTO 40
```

Für den PC-1360 ist eine Programmierung für die Übertragung nicht erforderlich.

■ SPEICHERINHALTE

A$	Für Ausgabe einer Programmzeile
FI$	Dateiname

■ ZEILENPLAN

Zeilennummer	Inhalt
10	Initialisierung
20	Eingabe des Dateinamen
30～80	Ausgabe zum RS-232C

Programmname: RUNDE GRAFIK

Dazu ist der Farb-Matrixdrucker CE-140P erforderlich.

■ ÜBERSICHT

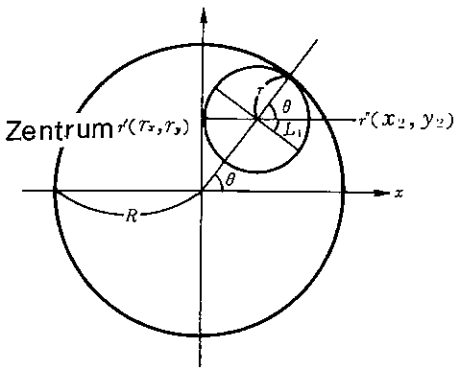
Mit diesem kurzen Programm können Sie verschiedene interessante Bögen in Farbe zeichnen.

■ ANWEISUNGEN

1. Das Programm wird mit **DEF** **A** gestartet.
2. Geben Sie die Farbe für die Grafik an.

0	1	2	3	4	5	6
SCHWARZ	PURPURN	ROT	MAGENTA	CRÜN	ZYAN	GELB

3. Geben sie L ein. L ist der Abstand zwischen dem Radianten des kleineren Kreises und dem Drehpunkt im Zentrum des kleineren Kreises.
4. Nach Abschluß der oben beschriebenen Eingaben wird eine Grafik ausgedruckt.



Wert r'' für Variable θ wird wie folgt erhalten:

$$x_2 = x_1\cos\theta 1 - y_1\sin\theta 1 + r_x$$

$$y_2 = x_1\sin\theta 1 + y_1\cos\theta 1 + ry$$

Beachte
$$\theta 1 = \theta - 90°$$
$$x_1 = L\cos(90° - \frac{R}{r}\theta)$$
$$y_1 = L\sin(90° - \frac{R}{r}\theta)$$
$$r_x = (R - r)\cos\theta$$
$$r_y = (R - r)\sin\theta$$

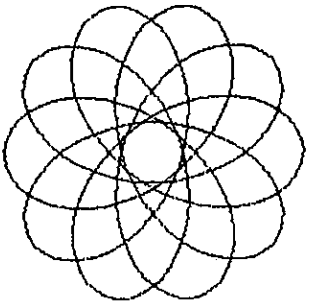
■ BEISPIEL

Erzeugen von Grafiken unter Verwendung nachfolgender Daten.

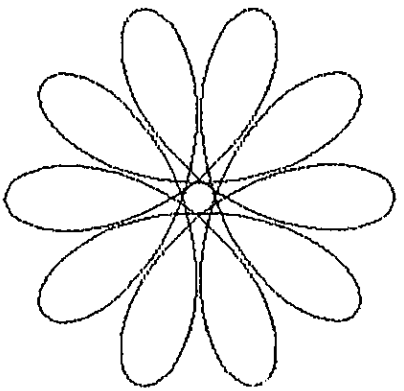
	Farbe	r	L
1	5	140	40
2	3	140	70

■ AUSDRUCK...(Wird in Farbe ausgedruckt)

BEISPIEL 1



BEISPIEL 2



■ PROGRAMMLISTE

```
 10:"A": WAIT 0: CLEAR : LPRINT :
    CONSOLE 39
 20:LPRINT : LTEXT : LPRINT : GRAPH :
    GLCURSOR (240,-240): SORGN
 30:CLS : USING : PRINT "  ** ROUND GRA
    PHIC **"
 40:INPUT "COLOR(0-6)=";M: COLOR M,7
 50:INPUT "r=";C,"L=";L
 60:R=200:H=10: CLS
 70:CURSOR 5,1: PRINT "** PRINTING **"
 80:P=10000
 90:D=0: GOSUB "CAL":U=O:V=Q:E=O:F=Q:
    GLCURSOR (U,V)
100:FOR I=H TO P STEP H
110:D=I: GOSUB "CAL":W=O:X=Q
120:LLINE -(W,X)
130:U=W:V=X: IF F=U IF E=U THEN 150
140:NEXT I
150:CLS : LTEXT : LPRINT : LF 10: END
160:"CAL":G=R/C*D:S=(R-C)* COS D:T=(R-C
    )* SIN D:K=L* COS (90-G):N=L* SIN (
    90-G)
170:A= SIN (D-90):B= COS (D-90)
180:O=K*B-A*N+S:Q=K*A+B*N+T
190:RETURN
```

453 bytes

■ SPEICHERINHALTE

C	r (RAdiant des kleineren Kreises)
D	θ
E	Erste X-Koordination
F	Erste Y-Koordination
G	θ'
H	$\sqrt{}$
J	θ_1
K	X_1
L	L
M	Farbeinstellung
N	Y_1
O	X_2
P	Grenzwert von O
Q	Y_2
R	R (Radiant des größeren Kreises)
T	r_X
T	r_Y

■ ZEILENPLAN

Zeilennummer	Inhalt
10 ~ 20	Initialsisierung
30 ~ 50	Dateneingabe
60 ~ 150	Ausdruck der Grafik
160 ~ 190	Berechnungs-Subprogramm

■ TASTENFOLGE

BEISPIEL 1

1) DEF A

```
  **ROUND GRAPHIC**
COLOR(Ø-6)= _
```

2) 5 ENTER

```
  **ROUND GRAPHIC**
COLOR (Ø-6)=5
r= _
```

3) 140 ENTER

```
**ROUND GRAPHIC**
 COLOR(Ø-6)= 5
 r=14Ø
 L= _
```

4) 40 ENTER

```
    ** PRINTING **
```

```
>
```

BEISPIEL 2

1) DEF A

```
  **ROUND GRAPHIC**
COLOR(Ø-6)= _
```

2) 3 ENTER

```
  **ROUND GRAPHIC**
COLOR (Ø-6)=3
r= _
```

3) 140 ENTER

```
**ROUND GRAPHIC**
 COLOR(Ø-6)= 3
 r=14Ø
 L= _
```

4) 70 ENTER

```
    ** PRINTING **
```

```
>
```

Programmname: SKISPRUNG

Wer springt am weitestesn?

Heute haben wir schönes Wetter, gerade richtig zum Skispringen. Sie sind zuversichtlich, den weitesten Sprung hinzulegen. Gehen Sie in den Anlauf, springen sie mit einem perfekten Timing und halten sie danach trotz Windstößen von links und rechts das Gleichgewicht.

Also, wollen wir einmal sehen, wieviele Meter Sie springen können!

■ WIE SPIELEN

1. Nach Drücken von **R** **U** **N** und **ENTER** wird "The longest distance" angezeigt. Drücken Sie die Leertaste und das Spiel beginnt.

2. Nachdem der Sprungstand auf der Anzeige angezeigt ist, startet der Mann. Wärend sich der Springer im anlauf befindet, drücken Sie die Taste **5** um ihn springen zu lassen. Springt der Springer aufgrund eines Fehlers im Timing nicht, wird "Om" angezeigt und der Spielablauf kehrt zur ursprünglichen Titelanzeige zurück.

3. Obwoho der Springer abgesprungen ist, müssen sie wegen der Seitenwinde mittels folgender Tasten das Gleichgewicht des Springers halten:

   Wind von links (→): Drücken Sie sie Taste **4**
   wind von rechts (←): Drücken Sie die Taste **6**

   (Eine senkechte Linie "|" auf dem Bildschirm zeigt die Balance des Springers an).

   - Wenn der Springer richtig landet, wird die Sprungweite angezeigt. Sollte er aber während des sprunges das Gleichgewicht verlieren, wird "Fallen" angezeigt.
     In diesem Fall erscheint keine Anzeige der Sprungweite.

     Wenn der oben geschilderte Vorgang abgeschlossen ist, erscheint wieder die ursprüngliche Anzeige. Geben sie das spiel wieder ein, sollten Sie es noch einmal versuchen wollen.

■ HINWEIS

<Wie die sprungweite errechnet wird>

1. Timing des Sprunges
   J (die anlaufgeschwindigkeit des Springers entsprechend dem Timing) wird mittels der X-Koordinate (125-110) bei Drücken der Taste **5** berechnet.

   $$J=(X-110)\times 0.02$$

   Der Springer bewegt sich entlang den Dots bis zu den X-Koordinaten 105-80. Mit jeder Bewegung fährt der Springer weiter nach unten aufgrund der Formel "Y=Y+J" (Y steht für Höhe).
   Kurz gesagt, der Zeitpunkt des Drückens der Taste **5** bestimmt die Höhe am Punkt der X-Koordinaten 80.

2. Gleichgewicht
   Die Höhe verringert sich gleichmäßig. Die X-Koordinaten werden mittels folgender Formel berechnet:

   $$X=X-(5-ABS(B))\times 0.2$$

   B zeigt das Gleichgewicht im Bereich zwischen -5 und $+5$ an.
   (Ist B über oder unter diesem Bereich, stürzt der Springer ab.)

   Die Berechnung und die Anzeige werden in oben beschriebener Reihenfolge wiederholt und sobald die Bewertungsformel für die Landung "Y>31−INT(X×0.05)" festent, wird S (Sprungweite) mittels der X-Koordinaten berechnet.

   $$S=(100-X)\times 1.5$$

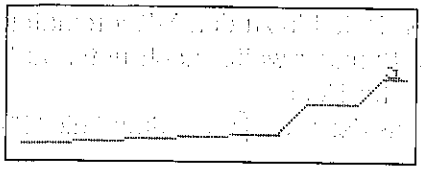
   Im halten der Balance während des Fluges besteht also der Trick für das Erreichen einer großen Weite.

■ TASTENFOLGE

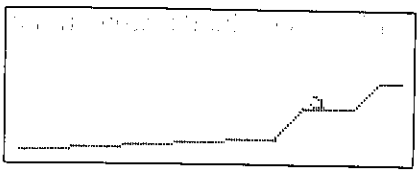
1) **R** **U** **N** **ENTER**
[Programmstart]

```
*** Ski Jump ***
The Longest Distance:
                    0m
Press the space key
```

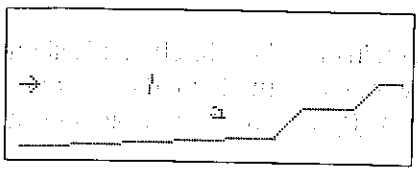
2) ▭ (Leertaste)
[Das Spiel beginnt]



3) **5**

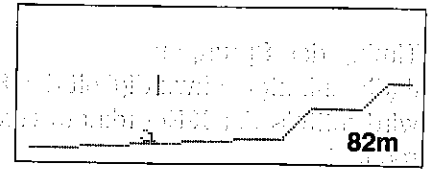


4) **4**

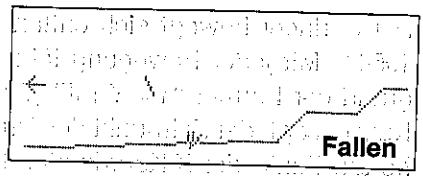


Betätigen der Taste 4 oder 6 setzt das Spiel fort.

«Anzeigebeispiel einer guten Landung»



«Anzeigebeispiel eines Absturzes»



■ PROGRAMMLISTE

```
10:WAIT 0:H=0
20:F1$="40888890F08000"
   :F2$="40888890F08000
   00":F3$="1EE05E20182
   4"
30:DIM S$(5)*16:S$(1)="
   030C30C000000000":S$
   (2)="00000FF00000000
   0"
40:S$(3)="000000FF00000
   0":S$(4)="000000F00F
   000000":S$(5)="00000
   000C0300C03"
50:CLS : WAIT 0:
   CURSOR 1,0: PRINT "*
   ** Ski Jump ***":
   BEEP 1
60:CURSOR 1,1: PRINT "T
   he Longest Distance
   :";
70:CURSOR 17,2: PRINT
   USING "####";H;"m";
80:CURSOR 1,3: PRINT "P
   ress the space key";
90:I$= INKEY$ : IF I$<>
   " " THEN 90
100:CLS : LINE (149,5)-(
   140,5): LINE -(130,1
   5): LINE -(110,15):
   LINE -(99,26)
110:FOR I=1 TO 5: LINE (
   119-I*20,26+I)-(100-
   I*20,26+I): NEXT I
120:GCURSOR (141,4):
   GPRINT F1$: GCURSOR
   (141,4): BEEP 3:
   GPRINT "000000000000
   "
130:FOR I=0 TO 10:
   GCURSOR (134-I,4+I):
   GPRINT F1$: NEXT I
140:X=125
150:I$= INKEY$ : IF I$="
   5" THEN 210
160:X=X-2: GCURSOR (X,14
   ): GPRINT F2$: IF X>
   110 THEN 150
170:GCURSOR (X,14):
   GPRINT "000000000000
   "
180:FOR I=0 TO 11:
   GCURSOR (103-I,15+I)
   : GPRINT F1$: NEXT I
190:CURSOR 19,3: WAIT 16
   0: PRINT "0 m";
200:GOTO 50
210:J=(X-110)*.02: FOR I
   =X TO 105 STEP -2:
   GCURSOR (I,14):
   GPRINT F2$: NEXT I
220:Y=14: FOR I=105 TO 8
   0 STEP -1:Y=Y+J:
   GCURSOR (I,Y):
   GPRINT F1$: NEXT I:B
   =0,X=80
230:R= RND 3: ON R GOTO
   240,250,260
240:GCURSOR (0,10):
   GPRINT "101010109254
   3810":B=B+1: GOTO 27
   0
250:GCURSOR (0,10):
   GPRINT "0000000000000
   0000": GOTO 270
260:GCURSOR (0,10):
   GPRINT "103854921010
   1010":B=B-1
270:I$= INKEY$
280:IF I$="4" LET B=B-1
290:IF I$="6" LET B=B+1
300:IF B<-5 OR B>5 THEN
   390
310:GCURSOR (50,10):
   GPRINT S$(B*.5+3)
320:X=X-(5- ABS (B))*.2:
   Y=Y+.26: IF Y>31-
   INT (X*.05) THEN 360
330:GCURSOR (X,Y):
   GPRINT F2$
340:IF Y>27 LET C=
   POINT (X,Y+1): IF C
   GOSUB 420
350:GOTO 230
360:S=(100-X)*1.5:
   CURSOR 18,3: PRINT
   USING "####";S;:
   WAIT 320: PRINT "m"
370:IF S>H LET H=S
380:GOTO 50
390:FOR I=Y TO 31:
   GCURSOR (X,I):
   GPRINT F3$: NEXT I
400:CURSOR 18,3: WAIT 16
   0: PRINT "Fallen";
410:GOTO 50
420:FOR I=2 TO 5: LINE (
   119-I*20,26+I)-(100-
   I*20,26+I): NEXT I:
   RETURN
```

1394 bytes

■ SPEICHERINHALT

B	Gleichgewicht
C	√
H	Größte Sprungweite
I,I$	Schleifenzähler √
J	Anlaufgeschwindigkeit
R	Windrichtung
S	Sprungweite
X	Niveauunterschied
Y	Höhe
F1$	Anzeige (Springer)
F2$	Anzeige (Springer)
F3$	Anzeige (Springer)
S$(5)	Anzeige

■ ZEILENPLAN

Zeilen-nummer	Inhalt
10 ~ 40	Initialisation
50 ~ 140	Anfangsanzeige
150 ~ 200	Sprungtiming
210 ~ 220	Sprungvorgang
230 ~ 260	Windrichting
270 ~ 350	Gleichgewicht
360 ~ 380	Landevorgang
390 ~ 410	Absturz
420	Anzeige der Bahn

INHALTSVERZEICHNIS

— A —

Abkürzungen 82
ALL RESET 10
Anweisungen 63, 110
Ausdrücke 56 , 58

— B —

Batterien 12
Befehle 64
Begriffe 44
Beispiele 339
Beschreibung 4

— C —

— D —

Display 8

— E —

— F —

Fehler 27
Formatieren 318
Funktionen 326

— G —

Grafik 111
Grafikbildschirm 76

— H —

Hexadezimalzahlen 45

— I —

— J —

— K —

Kommandos 65
Kopieren von RAM-Karten 107

— L —

Logische 58

— M —

Maschinensprache-Kommandos 112
Meldungen 311

— N —

Numerische Ausdrücke 56

— O —

— P —

Pannenhilfe 303
Peripehergeräte 87
Plotter/Druckergrafik 111
Programmieren 63

— R —

RAM-Karten 102
Rangfolge 41
RESERVE-Modus 83
RUN-Modus 61

— S —

Schablone 86
Serien-E/A 98
Speichern von Programmen 75
Strom ein/aus 19

— T —

Tastatur 4
Technische Daten 334

— U —

— V —

Variablen 45, 47, 48, 49, 53
Verhältnis 57

— W —

Wartung 310
Wiederabruf von Eingaben 23
Wissenschaftliche Notation 35, 321

Zeichen-
Codetabelle 315
Folgen 49, 57

SHARP CORPORATION

OSAKA, JAPAN

©1986 SHARP CORPORATION

PRINTED IN JAPAN/IMPRIMÉ AU JAPON

6G 10.0-I(TINSG1075ECZZ)